Planning with Attitude

Brian E Jackson[1], Kevin Tracy[1], and Zachary Manchester[1]

Abstract—Planning trajectories for floating-base robotic systems that experience large attitude changes is challenging due to the nontrivial group structure of 3D rotations. This paper introduces a powerful and accessible approach for optimization-based planning on the space of rotations using only standard linear algebra and vector calculus. We demonstrate the effectiveness of the approach by adapting Newton's method to solve the canonical Wahba's problem, and modify the trajectory optimization solver ALTRO to plan directly on the space of unit quaternions, achieving superior convergence on problems involving significant changes in attitude.

Index Terms—Motion and Path Planning, Optimization and Optimal Control, Computational Geometry, Underactuated Robots, Motion Control,

I. INTRODUCTION

MANY robotic systems—including quadrotors, airplanes, satellites, autonomous underwater vehicles, and quadrupeds—can perform arbitrarily large three-dimensional translations and rotations as part of their normal operation. While representing translations is straightforward and intuitive, effectively representing the nontrivial group structure of 3D rotations has been a topic of study for many decades. Although we can intuitively deduce that rotations are three-dimensional, a globally non-singular three-parameter representation of the space of rotations does not exist [1]. As a result, when parameterizing rotations, we must either a) choose a three-parameter representation and deal with singularities and discontinuities, or b) choose a higher-dimensional representation and deal with constraints between the parameters. While simply representing attitude is nontrivial, generating and tracking motion plans for floating-base systems is an even more challenging problem.

Early work on control problems involving the rotation group dates back to the 1970s, with extensions of linear control theory to spheres [2] and $SO(3)$ [3]. Effective attitude tracking controllers have been developed for satellites [4], quadrotors [5]–[10], and a 3D inverted pendulum [11] using various methods for calculating three-parameter attitude errors.

More recently, these ideas have been extended to trajectory generation [12], sample-based motion planning [13], [14], and optimal control. Approaches to optimal control with attitude states include analytical methods applied to satellites [15], discrete mechanics [16]–[18], a combination of sampling-based planning and constrained trajectory optimization for satellite formations [19], [20], projection operators [21], or more general theory for optimization on manifolds [22]. Nearly all of these methods rely heavily on principles from differential geometry and Lie group theory; however, despite these works, many recent papers in the robotics community continue to naively apply standard methods for motion planning and control with no regard for the group structure of rigid body motion.

In this paper, we make a departure from previous approaches to geometric planning and control that rely heavily on ideas and notation from differential geometry, and instead use only basic mathematical tools from linear algebra and vector calculus that should be familiar to most roboticists. In Sec. III we introduce an approach to quaternion differential calculus similar to [23], [24], but significantly simpler and more general, enabling straightforward adaptation of existing algorithms to systems with quaternion states. For concreteness, we then apply our method to the canonical Wahba's problem [25] in Sec. IV, and demonstrate superior convergence to approaches that fail to properly account for the group structure. In Sec. V we extend these ideas to the problem of trajectory optimization, and detail modifications to ALTRO, a state-of-the-art constrained trajectory optimization solver, and demonstrate performance gains on several benchmark problems. With the modifications presented in this paper, ALTRO explicitly leverages both the structure of the trajectory optimization problem as well as the group structure of 3D rotations, making it uniquely well-suited to solving challenging problems with near real-time performance.

In summary, our contributions include:

- A unified approach to quaternion differential calculus entirely based on standard linear algebra and vector calculus.
- Derivation of a Newton-based algorithm for nonlinear optimization directly on the space of unit quaternions.
- Implementation of a fast and efficient solver for trajectory optimization problems with attitude dynamics and nonlinear constraints that correctly accounts for the group structure of 3D rotations.

II. Background

We begin by defining some useful conventions and notation. Attitude is defined as the rotation from the robot's body frame to the world frame. We also define gradients to be row vectors, that is, for $f(x) : \mathbb{R}^n \to \mathbb{R}$, $\frac{\partial f}{\partial x} \in \mathbb{R}^{1\times n}$.

Manuscript received: October, 15, 2020; Revised January 2, 2021; Accepted January 4, 2021.

This paper was recommended for publication by Editor Nancy Amato upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by a NASA Early Career Faculty Award, NASA JPL, and NSF GFRP Grant No. DGE-1656518.

[1]All authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA brianjackson@cmu.edu

Digital Object Identifier (DOI): see top of this page.

A. Unit Quaternions

We leverage the fact that quaternions are linear operators and that the space of quaternions $\mathbb{H}$ is isomorphic to $\mathbb{R}^4$ to explicitly represent—following the Hamilton convention—a quaternion $\mathrm{q} \in \mathbb{H}$ as a standard vector $q \in \mathbb{R}^4 := [q_s \quad q_v^T]^T$ where $q_s \in \mathbb{R}$ and $q_v \in \mathbb{R}^3$ are referred to as the scalar and vector parts of the quaternion, respectively. The space of unit quaternions, $\mathbb{S}^3 = \{q : \|q\|_2 = 1\}$, is a double-cover of the rotation group $SO(3)$, since q and $-q$ represent the same rotation [26].

Quaternion multiplication is defined as

$$\mathrm{q}_2 \otimes \mathrm{q}_1 = L(q_2)q_1 = R(q_1)q_2 \tag{1}$$

where $L(q)$ and $R(q)$ are orthonormal matrices defined as

$$L(q) := \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I + [q_v]^\times \end{bmatrix} \tag{2}$$

$$R(q) := \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I - [q_v]^\times \end{bmatrix}, \tag{3}$$

and $[x]^\times$ is the skew-symmetric matrix operator

$$[x]^\times := \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{4}$$

The inverse of a unit quaternion q^{-1}, giving the opposite rotation, is equal to its conjugate q^*, which is simply the same quaternion with a negated vector part:

$$\mathrm{q}^* = Tq := \begin{bmatrix} 1 & \\ & -I_3 \end{bmatrix} q. \tag{5}$$

The following identities, which are easily derived from (2)–(5), are extremely useful:

$$L(Tq) = L(q)^T = L(q)^{-1} \tag{6}$$

$$R(Tq) = R(q)^T = R(q)^{-1}. \tag{7}$$

We will sometimes find it helpful to create a quaternion with zero scalar part from a vector $r \in \mathbb{R}^3$. We denote this operation as,

$$\hat{r} = Hr \equiv \begin{bmatrix} 0 \\ I_3 \end{bmatrix} r. \tag{8}$$

Unit quaternions rotate a vector through the operation $\hat{r}' = \mathrm{q} \otimes \hat{r} \otimes \mathrm{q}^*$. This can be equivalently expressed using matrix multiplication as

$$r' = H^T L(q) R(q)^T H r = A(q) r, \tag{9}$$

where $A(q)$ is the rotation matrix in terms of the elements of the quaternion [27].

B. Rigid Body Dynamics

For clarity, we will restrict our attention to rigid bodies moving freely in 3D space. That is, we consider systems with dynamics of the following form:

$$x = \begin{bmatrix} r \\ q \\ v \\ \omega \end{bmatrix}, \quad \dot{x} = \begin{bmatrix} v \\ \frac{1}{2}\mathrm{q} \otimes \hat{\omega} = \frac{1}{2} L(q) H \omega \\ \frac{1}{m} {}^W F(x,u) \\ J^{-1}({}^B\tau(x,u) - \omega \times J\omega) \end{bmatrix} \tag{10}$$

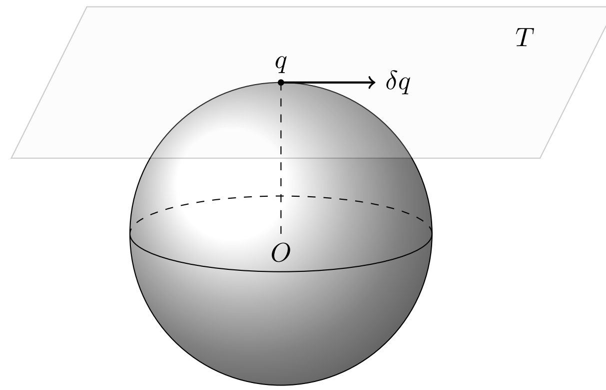

Fig. 1. When linearizing about a point q on an sphere $\mathbb{S}^{n-1}$ in n-dimensional space, the tangent space T is a plane living in $\mathbb{R}^{n-1}$, illustrated here with $n = 3$. Therefore, when linearizing about a unit quaternion $q \in \mathbb{S}^3$, the space of differential rotations lives in $\mathbb{R}^3$.

where x and u are the state and control vectors, $r \in \mathbb{R}^3$ is the position, $\mathrm{q} \in \mathbb{S}^3$ is the attitude, $v \in \mathbb{R}^3$ is the linear velocity, and $\omega \in \mathbb{R}^3$ is the angular velocity. $m \in \mathbb{R}$ is the mass, $J \in \mathbb{R}^{3\times3}$ is the inertia matrix, ${}^W F(x,u) \in \mathbb{R}^3$ are the forces in the world frame, and ${}^B\tau(x,u)$ are the moments in the body frame.

III. Quaternion Differential Calculus

We now present a simple but powerful method for taking derivatives of functions involving quaternions based on the notation and linear algebraic operations outlined in Sec. II-A.

Derivatives consider the effect an infinitesimal perturbation to the input has on an infinitesimal perturbation to the output. For vector spaces, the composition of the perturbation with the nominal value is simple addition and the infinitesimal perturbation lives in the same space as the original vector. For unit quaternions, however, neither of these are true; instead, they compose according to (1), and infinitesimal unit quaternions are (to first order) confined to a 3-dimensional plane tangent to $\mathbb{S}^3$ (see Fig. 1).

The fact that differential unit quaternions are three-dimensional should make intuitive sense: Rotations are inherently three-dimensional and differential rotations should live in the same space as angular velocities, i.e. $\mathbb{R}^3$.

There are many possible three-parameter representations for small rotations in the literature. Many authors use the exponential map [3], [12], [18], [21], [22], [28], [29], while others have used the Cayley map (also known as Rodrigues parameters) [16], [17], Modified Rodrigues Parameters (MRPs) [30], or the vector part of the quaternion [5]. We choose Rodrigues parameters [26] because they are computationally efficient and do not inherit the sign ambiguity associated with unit quaternions. The mapping between a vector of Rodrigues parameters $\phi \in \mathbb{R}^3$ and a unit quaternion q is known as the Cayley map:

$$q = \varphi(\phi) = \frac{1}{\sqrt{1 + \|\phi\|^2}} \begin{bmatrix} 1 \\ \phi \end{bmatrix}. \tag{11}$$

We will also make use of the inverse Cayley map:

$$\phi = \varphi^{-1}(q) = \frac{q_v}{q_s}. \tag{12}$$

A. Jacobian of Vector-Valued Functions

When taking derivatives with respect to quaternions, we must take into account both the composition rule and the nonlinear mapping between the space of unit quaternions and our chosen three-parameter error representation.

Let $\phi \in \mathbb{R}^3$ be a differential rotation applied to a function with quaternion inputs $y = h(q) : \mathbb{S}^3 \to \mathbb{R}^p$, such that

$$y + \delta y = h(L(q)\varphi(\phi)) \approx h(q) + \nabla h(q)\phi. \tag{13}$$

Note that we chose to represent ϕ in the body frame, consistent with the standard definition of angular velocity, and therefore it is applied to q through right (rather than left) multiplication. We can calculate the Jacobian $\nabla h(q) \in \mathbb{R}^{p\times 3}$ by differentiating (13) with respect to ϕ, evaluated at $\phi = 0$:

$$\nabla h(q) = \frac{\partial h}{\partial q} L(q) H := \frac{\partial h}{\partial q} G(q) = \frac{\partial h}{\partial q} \begin{bmatrix} -q_v^T \\ q_s I_3 + [q_v]^\times \end{bmatrix} \tag{14}$$

where $G(q) \in \mathbb{R}^{4\times 3}$ is the attitude Jacobian, which essentially becomes a "conversion factor" allowing us to apply results from standard vector calculus to the space of unit quaternions. This form is particularly useful in practice since $\partial h/\partial q \in \mathbb{R}^{p\times 4}$ can be obtained using finite differences or automatic differentiation. As an aside, although we have used Rodrigues parameters, $G(q)$ is actually the same (up to a constant scalar factor) for any choice of three-parameter attitude representation.

B. Hessian of Scalar-Valued Functions

If the output of h is a scalar ($p = 1$), then we can find its Hessian by taking the Jacobian of (14) with respect to ϕ using the product rule, again evaluated at $\phi = 0$:

$$\nabla^2 h(q) = G(q)^T \frac{\partial^2 h}{\partial q^2} G(q) - I_3 \frac{\partial h}{\partial q} q, \tag{15}$$

where the second term comes from the second derivative of $\varphi(\phi)$. Similar to $G(q)$, this expression is the same (up to a constant scalar factor) for any choice of three-parameter attitude representation.

C. Jacobian of Quaternion-Valued Functions

We now consider the case of a function that maps unit quaternions to unit quaternions, $q' = f(q) : \mathbb{S}^3 \to \mathbb{S}^3$. Here we must also consider the non-trivial effect of a differential rotation applied to the output, i.e.:

$$L(q')\varphi(\phi') = f(L(q)\varphi(\phi)). \tag{16}$$

Solving (16) for ϕ' we find,

$$\phi' = \varphi^{-1}\left(L(q')^T f(L(q)\varphi(\phi))\right) \approx \nabla f(q)\,\phi. \tag{17}$$

Finally, the desired Jacobian is obtained by taking the derivative of (17) with respect to ϕ:

$$\nabla f(q) = H^T L(q')^T \frac{\partial f}{\partial q} L(q) H = G(q')^T \frac{\partial f}{\partial q} G(q). \tag{18}$$

Once again, (18) holds (up to a constant) for any three-parameter attitude representation.

IV. Modifying Newton's Method

Newton's method uses derivative information about a function to iteratively approximate its roots. For unconstrained systems, this method is highly effective, and can exhibit quadratic convergence rates. For constrained systems, the updates can be projected back onto the feasible set at each iteration, but without the same convergence guarantees.

In this section, we will leverage the quaternion calculus results introduced in the previous section to modify Newton's method so that it implicitly accounts for the quaternion unit-norm constraint. Unlike the projection approach, this modified form of Newton's method retains the fast convergence rates associated with the unconstrained method. We will demonstrate this behavior on Wahba's Problem, a least-squares attitude estimation problem [25], [26].

A. Methodology

Given a set of known vectors in the world frame, ${}^W w_i$, and measurements of these vectors in the body frame, ${}^B v_i$, we seek the rotation from the body to the world frame ${}^W A(q)^B$ that solves the following optimization problem,

$$\begin{aligned} \underset{q}{\text{minimize}} \quad & W(q) \\ \text{subject to} \quad & q \in \mathbb{S}^3, \end{aligned}$$

where Wahba's loss function $W(q)$ is,

$$W(q) = \sum_i \left\| {}^W w_i - A(q)\ {}^B v_i \right\|_2^2 = \|r(q)\|_2^2, \tag{19}$$

and $r(q)$ is the residual vector.

The Jacobian of $r(q)$ can be found using (14):

$$\nabla r(q) = \frac{\partial r}{\partial q} G(q) \tag{20}$$

$$= -2H^T R(q)^T \left(\sum_i R({}^B \hat{v}_i) \right) G(q). \tag{21}$$

Given a guess solution, q_k, The standard Gauss-Newton method can then be used to compute a three-parameter update, ϕ_k via the Moore-Penrose psuedoinverse:

$$\phi_k = -(\nabla r^T \nabla r)^{-1} \nabla r^T r(q_k). \tag{22}$$

The update is then applied using the composition for the group:

$$\mathrm{q}_{k+1} = \mathrm{q}_k \otimes \varphi(\phi_k). \tag{23}$$

This "multiplicative" Gauss-Newton method is summarized in Algorithm 1.

Algorithm 1 Multiplicative Gauss-Newton Method

```
k = 0
while ||φ|| > tolerance do
    ∇r = (∂r(q_k)/∂q) G(q_k)                ▷ Compute Jacobian
    φ = −(∇r^T ∇r)^{−1} ∇r^T r(q_k)         ▷ Compute update step
    q_{k+1} = L(q_k) φ(φ)                   ▷ Apply update step
    k = k + 1
end while
```

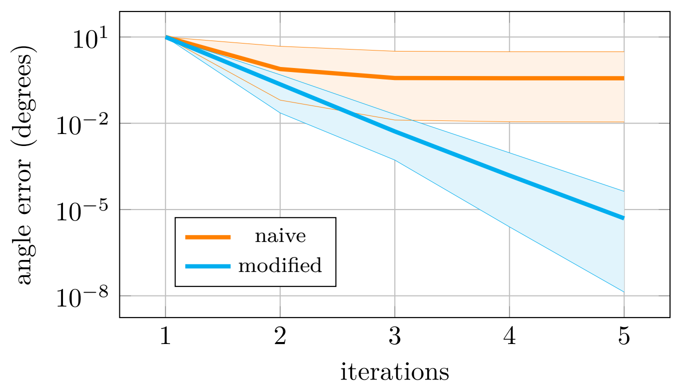

Fig. 2. Convergence comparison for Wahba's problem. The error is the angle between the current solution and the globally optimal solution computed using a singular-value decomposition. The thick line is the average result of 100 trials with randomized orientations and measurements. The thin lines are the maximum and minimum over all 100 trials. By modfying Newton's method with the methods of section III, quadratic convergence rates are achieved, while a naïve approach stalls after only a few iterations.

B. Results

Figure 2 compares the multiplicative Gauss-Newton method with a naïve Newton's method in which the quaternion is simply projected back onto the unit sphere at every iteration. The naïve method makes progress initially, but quickly stalls. By correctly handling the group structure of unit quaternions, the multiplicative method is able to maintain the fast convergence rates typical of Newton's method. By comparing our method with the global solution obtained from a singular-value decomposition [31], we see that our method recovers the globally optimal solution within a small number of iterations.

V. Trajectory Optimization for Rigid Bodies

Here we outline the modifications to the ALTRO solver [32] to solve trajectory optimization problems for rigid bodies, which extends easily to arbitrary systems whose state is in $\mathbb{R}^n \times \mathbb{S}^3$. We consider trajectory optimization problems of the form,

$$\begin{aligned} \underset{x_{1:N},\, u_{1:N-1}}{\text{minimize}} \quad & \ell_f(x_N) + \sum_{k=1}^{N-1} \ell_k(x_k, u_k) \\ \text{subject to} \quad & x_{k+1} = f(x_k, u_k), \\ & g_k(x_k, u_k) \le 0, \\ & h_k(x_k, u_k) = 0, \end{aligned} \tag{24}$$

where x and u are the state and control vectors as described in Sec. II-B, f are the dynamics as defined in (10), ℓ_k is a general nonlinear cost function at a single time step, N is the number of time steps, and g_k, h_k are general nonlinear inequality and equality constraints.

ALTRO combines techniques from both differential dynamic programming (DDP) and direct transcription methods to achieve high performance on challenging constrained nonlinear trajectory optimization problems. Like most methods for nonlinear optimization, ALTRO iteratively approximates the nonlinear functions f, ℓ, g, and h with their first or second-order Taylor series expansions. Leveraging the methods from Sec: III, we adapt the algorithm to optimize directly on the error state $\delta x \in \mathbb{R}^{12}$:

$$\delta x_k = \begin{bmatrix} r_k - \bar{r}_k \\ \varphi^{-1}(\bar{\mathrm{q}}_k^{-1} \otimes \mathrm{q}_k) \\ v_k - \bar{v}_k \\ \omega_k - \bar{\omega}_k \end{bmatrix}. \tag{25}$$

We begin by linearizing the dynamics about the reference state and input trajectories, $\bar{x}$ and $\bar{u}$, using (18). The linearized error dynamics become,

$$\delta x_{k+1} = A_k \delta x_k + B_k \delta u_k, \tag{26}$$

where

$$\begin{aligned} A_k &= E(\bar{x}_{k+1})^T \frac{\partial f}{\partial x}\Big|_{\bar{x}_k, \bar{u}_k} E(\bar{x}_k), \\ B_k &= E(\bar{x}_{k+1})^T \frac{\partial f}{\partial u}\Big|_{\bar{x}_k, \bar{u}_k}, \end{aligned} \tag{27}$$

and $E(x_k) \in \mathbb{R}^{13 \times 12}$ is the error-state Jacobian:

$$E(x) = \begin{bmatrix} I_3 & & & \\ & G(q) & & \\ & & I_3 & \\ & & & I_3 \end{bmatrix}. \tag{28}$$

By applying (14) and (15) to the nonlinear cost functions ℓ and (18) to the nonlinear constraint functions g_k and h_k, we can calculate the second-order expansion of the cost function:

$$\begin{aligned} \delta \ell(x, u) \approx \frac{1}{2} \delta x^T \ell_{xx} \delta x + \frac{1}{2} \delta u^T \ell_{uu} \delta u + \delta_u^T \ell_{ux} \delta x \\ + \ell_x^T \delta x^T + \ell_u^T \delta u. \end{aligned} \tag{29}$$

With these results, we can apply standard Newton and quasi-Newton techniques along the lines of Section IV. We can also calculate a second-order expansion of the "action-value function" $Q(x, u)$ needed in DDP and LQR-based methods,

$$Q_{xx} = \ell_{xx} + A_k^T P_{k+1} A_k \tag{30}$$

$$Q_{uu} = \ell_{uu} + B_k^T P_{k+1} B_k \tag{31}$$

$$Q_{ux} = \ell_{ux} + B_k^T P_{k+1} A_k \tag{32}$$

$$Q_x = \ell_x + A_k^T p_{k+1} \tag{33}$$

$$Q_u = \ell_u + B_k^T p_{k+1}, \tag{34}$$

from which we can calculate the quadratic expansion of the cost-to-go $P_k \in \mathbb{R}^{12 \times 12}$, $p_k \in \mathbb{R}^{12}$, and optimal linear feedback gains $K_k \in \mathbb{R}^{m \times 12}$ and feed-forward corrections

$d_k \in \mathbb{R}^m$ by starting at the terminal state and performing a backward Riccati recursion as usual [32], [33].

During the "forward rollout" of these methods, the dynamics are simulated forward in time using updated control inputs:

$$u_k = \bar{u}_k - d_k - K_k \delta x_k. \tag{35}$$

where $\bar{u}_k$ is the control value from the previous iteration, and δx is computed using (25). For more details on the ALTRO algorithm, we refer the reader to [32].

A. Quaternion Cost Functions

In addition to the straight-forward modifications to the ALTRO algorithm itself, some care must be taken in designing cost functions that are well-suited to unit quaternions. We frequently minimize costs that penalize distance from a goal state, e.g. $\frac{1}{2}(x - x_g)^T Q(x - x_g)$; however, naïve substraction of unit quaternions does not respect their group structure, and often results in undesired behavior. Instead, we have found the following cost function, which penalizes the geodesic distance between two unit quaternions [14], to work well in practice:

$$J_{\text{geo}} = (1 - |q_g^T q|). \tag{36}$$

Its gradient and Hessian are,

$$\nabla J_{\text{geo}} = -\operatorname{sign}(q_g^T q) q_g^T G(q) \tag{37}$$
$$\nabla^2 J_{\text{geo}} = \operatorname{sign}(q_g^T q) I_3 q_g^T q, \tag{38}$$

where sign denotes the signum function. This cost function is particularly useful for rotations since it eliminates the ambiguity arising from the quaternion double-cover of $SO(3)$.

VI. Experiments

In this section we present several trajectory optimization problems for systems that undergo large changes in attitude: an airplane barrel roll, a quadrotor flip, and a satellite with flexible solar panels that must slew to a new orientation while avoiding a keep-out zone. All problems are run using ALTRO, first without any of the modifications presented in the current paper, analagous to the naïve Newton's method in section IV and labeled "naive", and then using the modifications listed in Sec. V and the geodesic cost function described in Sec. V-A, labeled "modified". All cost functions are of the following form:

$$\ell_{\text{naive}}(x, u, \bar{x}, Q, R) = \frac{1}{2}(x - \bar{x})^T Q (x - \bar{x}) + \frac{1}{2} u^T R u \tag{39}$$

$$\ell_{\text{modified}}(x, u, \bar{x}, Q, R) = \ell_{\text{naive}}(x, u, \bar{x}, \bar{Q}, R) + w(1 \pm \bar{q}^T q) \tag{40}$$

where $\bar{x}$ is the reference state and $\bar{Q} = \operatorname{diag}(Q_r, \vec{0}_4, Q_v, Q_\omega)$, with Q_r, Q_v, Q_ω being the weights of Q for position, and linear and angular velocity, respectively.

TABLE I
Trajectory Optimization Timing Results (naive/modified)

Problem	Iterations	time (ms)
barrellroll	23 / 17	105.74 / 72.64
quadflip	31 / 25	505.59 / 433.59
satellite	16 / 17	170.98 / 263.10

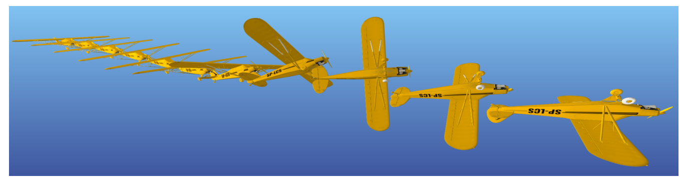

Fig. 3. Barrel roll trajectory computed by ALTRO using a terminal cost to encourage an upside-down attitude.

Timing results are summarized in Table I. All experiments are solved to a constraint satisfaction tolerance of 10^{-5} and discretized with a 4th order Runge-Kutta integrator. The results were run on a laptop computer with a 2.8 GHz i7-1165G7 processor with 16 GB of RAM. Code for all experiments is available on GitHub[1].

A. Airplane Barrel Roll

A 180 degree barrel roll trajectory for a fixed-wing airplane was optimized. The airplane's dynamics model consists of the a simple rigid body as defined in Section II-B with forces and torques due to lift and drag fit from wind tunnel data [34]. The airplane was tasked to do a barrel roll by constraining the terminal state to upside-down (see Fig. 3). To mitigate issues with integration error and drift in the magnitude of the quaternion, the following constraint function was used to enforce a terminal orientation of $\bar{q}$:

$$\frac{q_v}{\|q\|} - \bar{q}_v \operatorname{sign}\left(\bar{q}^T \frac{q}{\|q\|}\right) = 0 \tag{41}$$

The solver was initialized with level flight trim conditions. The convergence of the different versions of ALTRO is compared in Fig. 4. As expected, the modified version achieves better convergence and faster solve times compared to the naïve version since the expansions being provided to the algorithm more accurately capture the relationship between the attitude state and the goal and constraints. For this relatively simple problem, we gained a modest 31% improvement in runtime, despite the additional matrix multiplications when calculating the cost and constraint expansions.

B. Quadrotor Flip

A 360 degree flip trajectory for a quadrotor was optimized with dynamics adapted from [35]. To encourage

[1] https://github.com/RoboticExplorationLab/PlanningWithAttitude

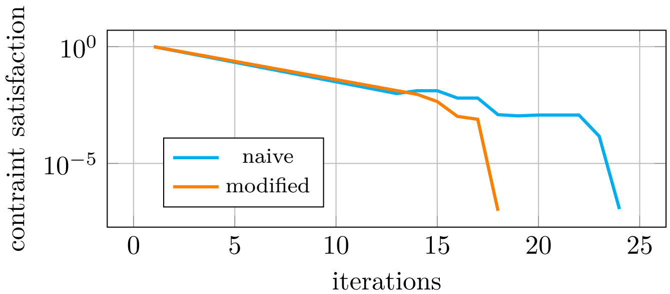

Fig. 4. Constraint satisfaction as a function of iteration when solving the barrel roll problem using ALTRO both with and without the modifications for optimizing unit quaternions.

the flip, we specified a "waypoint" cost function of the following form:

$$\sum_{k \in \mathcal{N}} \ell(x_k, u_k, \hat{x}, \hat{Q}, R) \; + \sum_{k \in \mathcal{W}} \ell(x_k, u_k, \bar{x}_k, Q_w, R) \tag{42}$$

where $\hat{x}$, $\hat{Q}$ are the nominal state and state weight matrix, Q_w is the weight matrix for the waypoints, and $\mathcal{W} = \{20, 45, 51, 55, 75, 101\}$, $\mathcal{N} = \{1 : 101\} \setminus \mathcal{W}$. Four intermediary "waypoints" were used to encourage the quadrotor to reach angles of 90°, 180°, 270°, and 360° around an approximately circular arc. The last waypoint was used to encourage the quadrotor to move towards the final goal, and the first kept it above the floor before starting the loop. The solver was provided a dynamically infeasible initial trajectory that linearly interpolates between the initial and final states, rotating the quadrotor around the x-axis a full 360°.

Figure 5 shows snapshots of the trajectory as generated using ALTRO. To compare the convergence properties of the two methods, the optimal state and control trajectories were perturbed with random Gaussian white noise with a mean of 1 for position, linear velocity, and angular velocity, 0.1 for the controls, and 145 degrees for the orientation (about a random axis). As shown in Fig. 6, the modified method converges more reliably than the naïve method. It is also worth noting that this problem could not be solved using any three-parameter attitude representation, since it passes through the singularities at $90°, 180°$, and $360°$ associated with Euler angles, Rodrigues parameters, and Modified Rodrigues Parameters, respectively.

C. Satellite Attitude Keep-Out

A spacecraft with flexible appendages was tasked to perform a 150 degree slew maneuver while ensuring that a body-mounted camera did not point within 40 degrees of a "keep-out zone" around the sun vector. The spacecraft dynamics are presented in detail in [36], and are based on equation (10) with the addition of six states to account for three flexible modes. Control torques are generated by four reaction wheels. A quadratic cost function penalizes error from the desired final attitude as well as displacement of

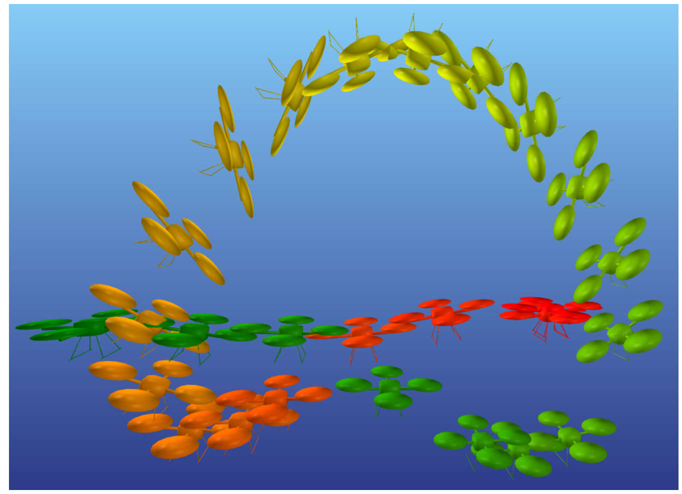

Fig. 5. Snapshots of the quadrotor flip trajectory. The gree-colored quadrotors represent the state near t=0 s and the red-colored quadrotors represent the state near t=5.0 s

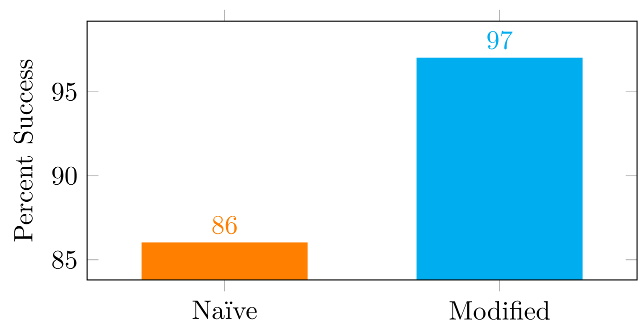

Fig. 6. Convergence comparison for quadrotor flip. Percent of 100 trials that successfully converged, where each trial is initialized with locally-optimal trajectories perturbed with significant Gaussian white noise.

the flexible modes. We enforce the camera keep-out zone with the following constraint,

$$\left({}^{W}r_{sun}\right)^{T} \left({}^{W}A(q)^{B}\,{}^{B}r_{cam}\right) \leq \cos(40°), \tag{43}$$

where ${}^{B}r_{cam}$ is the camera line-of-sight unit vector in the body frame and ${}^{W}r_{sun}$ is the unit vector pointing to the sun in the world frame. The attitudes that satisfy this constraint comprise a non-convex set, with the constraint itself being nonlinear in q.

ALTRO is able to converge to a locally optimal trajectory for this problem without an initial guess (all controls were initialized to zero). The resulting attitude trajectory is depicted in Fig. 7. Without enforcing the camera constraint, the trajectory passes through the keep-out zone. As noted in Table I, the quaternion modifications did not result in a significant improvement over the naive implementation of ALTRO for this problem, indicating that the computational benefits are problem dependent. We hypothesize that the more dynamic behaviors in the other examples benefit more from the quaternion modifications than the relatively slow-moving spacecraft.

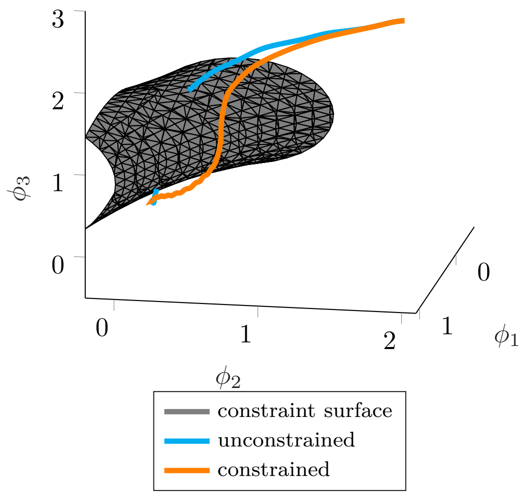

Fig. 7. Visualization of the flexible spacecraft slew with a keep-out zone. Attitude is parameterized with a Rodrigues parameter to visualize the trajectory in three dimensions. The constraint surface represents attitudes where the camera line-of-sight is within 40° of the sun. The unconstrained solution violates this constraint, while the constrained solution is able to avoid the keep out zone.

VII. Conclusions

We have presented a general, unified method for optimization-based planning and control for rigid-body systems with arbitrary attitude using standard linear algebra and vector calculus. The application of this methodology is straightforward and yields substantial improvements in the convergence of Newton and DDP-based methods, while also offering improvements for nonlinear constrained trajectory optimization for floating-base systems.

Many state-of-the-art trajectory optimization methods, including direct collocation and sequential convex programming, rely on general-purpose optimization solvers whose internal numerical methods are not exposed to the user. Therefore, these methods are unable to exploit the full structure of both the trajectory optimization problem and the rotation group at a low level. In contrast, we are able to implement deep, native support for quaternions into the ALTRO solver, making it possible to solve more challenging problems with higher performance than other algorithms.

In future work, we plan to apply ALTRO to real-time model-predictive control problems for aerial vehicles like quadrotors and airships that experience large attitude changes. The methods we have presented can also be leveraged to adapt other classes of gradient or Newton-based algorithms to exploit the structure of 3D rotations. Future directions beyond trajectory optimization may include simulation and planning methods that leverage maximal-coordinate formulations of multi-body dynamics [37], system-identification for complex multi-body systems and fixed-wing aircraft in post-stall conditions, and state estimation for spacecraft with sparse measurements.

Acknowledgements

This work was supported by a NASA Early Career Faculty Award (Grant Number 80NSSC18K1503). This research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration and funded through JPL's Strategic University Research Partnerships (SURP) program. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1656518. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] J. Stuelpnagel, "On the parametrization of the three-dimensional rotation group," SIAM review, vol. 6, no. 4, pp. 422–430, 1964.

[2] R. W. Brockett, "Lie Theory and Control Systems Defined on Spheres," en, SIAM Journal on Applied Mathematics, vol. 25, no. 2, pp. 213–225, Sep. 1973, issn: 0036-1399, 1095-712X. doi: 10.1137/0125025. [Online]. Available: http://epubs.siam.org/doi/10.1137/0125025 (visited on 01/07/2020).

[3] J. Baillieul, "Geometric methods for nonlinear optimal control problems," en, Journal of Optimization Theory and Applications, vol. 25, no. 4, pp. 519–548, Aug. 1978, issn: 0022-3239, 1573-2878. doi: 10.1007/BF00933518. [Online]. Available: http://link.springer.com/10.1007/BF00933518 (visited on 01/07/2020).

[4] B. Wie and P. M. Barba, "Quaternion feedback for spacecraft large angle maneuvers," Journal of Guidance, Control, and Dynamics, vol. 8, no. 3, pp. 360–365, 1985.

[5] E. Fresk and G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor," in 2013 European Control Conference (ECC), Jul. 2013, pp. 3864–3869. doi: 10.23919/ECC.2013.6669617.

[6] H. Liu, X. Wang, and Y. Zhong, "Quaternion-Based Robust Attitude Control for Uncertain Robotic Quadrotors," IEEE Transactions on Industrial Informatics, vol. 11, no. 2, pp. 406–415, Apr. 2015, issn: 1941-0050. doi: 10.1109/TII.2015.2397878.

[7] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in 49th IEEE conference on decision and control (CDC), IEEE, 2010, pp. 5420–5425.

[8] E. N. Johnson and S. K. Kannan, "Adaptive Trajectory Control for Autonomous Helicopters," en, Journal of Guidance, Control, and Dynamics, vol. 28, no. 3, pp. 524–538, May 2005, issn: 0731-5090, 1533-3884. doi: 10.2514/1.6271. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.6271 (visited on 01/07/2020).

[9] M. Watterson and V. Kumar, "Control of quadrotors using the hopf fibration on so (3)," in Robotics Research, Springer, 2020, pp. 199–215.

[10] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 2520–2525.

[11] N. A. Chaturvedi, N. H. McClamroch, and D. S. Bernstein, "Asymptotic Smooth Stabilization of the Inverted 3-D Pendulum," IEEE Transactions on Automatic Control, vol. 54, no. 6, pp. 1204–1215, Jun. 2009, issn: 2334-3303. doi: 10.1109/TAC.2009.2019792.

[12] M. Zefran, V. Kumar, and C. Croke, "On the generation of smooth three-dimensional rigid body motions," IEEE Transactions on Robotics and Automation, vol. 14, no. 4, pp. 576–589, Aug. 1998, issn: 2374-958X. doi: 10.1109/70.704225.

[13] M. Žefran, V. Kumar, and C. Croke, "Metrics and Connections for Rigid-Body Kinematics," en, The International Journal of Robotics Research, vol. 18, no. 2, pp. 243–258, Feb. 1999, issn: 0278-3649.

[14] J. Kuffner, "Effective sampling and distance metrics for 3D rigid body path planning," en, in IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, New Orleans, LA, USA: IEEE, 2004, 3993–3998 Vol.4, isbn: 978-0-7803-8232-9. doi: 10.1109/ROBOT.2004.1308895. [Online]. Available: http://ieeexplore.ieee.org/document/1308895/ (visited on 01/08/2020).

[15] K. Spindler, "Optimal control on Lie groups with applications to attitude control," en, Mathematics of Control, Signals, and Systems, vol. 11, no. 3, pp. 197–219, Sep. 1998, issn: 0932-4194, 1435-568X. doi: 10.1007/BF02741891. [Online]. Available: http://link.springer.com/10.1007/BF02741891 (visited on 01/07/2020).

[16] M. B. Kobilarov and J. E. Marsden, "Discrete Geometric Optimal Control on Lie Groups," IEEE Transactions on Robotics, vol. 27, no. 4, pp. 641–655, Aug. 2011, issn: 1941-0468. doi: 10.1109/TRO.2011.2139130.

[17] M. Kobilarov, "Discrete optimal control on lie groups and applications to robotic vehicles," in 2014 IEEE International Conference on Robotics and Automation (ICRA), May 2014, pp. 5523–5529. doi: 10.1109/ICRA.2014.6907671.

[18] T. Lee, M. Leok, and N. H. McClamroch, "Optimal Attitude Control of a Rigid Body Using Geometrically Exact Computations on SO(3)," en, Journal of Dynamical and Control Systems, vol. 14, no. 4, pp. 465–487, Oct. 2008, issn: 1079-2724, 1573-8698. doi: 10.1007/s10883-008-9047-7. [Online]. Available: http://link.springer.com/10.1007/s10883-008-9047-7 (visited on 12/19/2019).

[19] I. Garcia and J. How, "Trajectory optimization for satellite reconfiguration maneuvers with position and attitude constraints," in Proceedings of the 2005, American Control Conference, 2005., Jun. 2005, 889–894 vol. 2. doi: 10.1109/ACC.2005.1470072.

[20] G. S. Aoude, J. P. How, and I. M. Garcia, "Two-stage path planning approach for solving multiple spacecraft reconfiguration maneuvers," en, The Journal of the Astronautical Sciences, vol. 56, no. 4, pp. 515–544, Dec. 2008, issn: 0021-9142. doi: 10.1007/BF03256564. [Online]. Available: http://link.springer.com/10.1007/BF03256564 (visited on 01/08/2020).

[21] A. Saccon, J. Hauser, and A. P. Aguiar, "Optimal Control on Lie Groups: The Projection Operator Approach," IEEE Transactions on Automatic Control, vol. 58, no. 9, pp. 2230–2245, Sep. 2013, issn: 2334-3303. doi: 10.1109/TAC.2013.2258817.

[22] M. Watterson, S. Liu, K. Sun, T. Smith, and V. Kumar, "Trajectory optimization on manifolds with applications to $SO(3)$ and $\mathbb{R}^3 \times S^2$.," in Robotics: Science and Systems (RSS), Pittsburgh, PA, USA, Jun. 2018.

[23] D. P. Mandic, C. Jahanchahi, and C. C. Took, "A Quaternion Gradient Operator and Its Applications," IEEE Signal Processing Letters, vol. 18, no. 1, pp. 47–50, Jan. 2011, issn: 1558-2361. doi: 10.1109/LSP.2010.2091126.

[24] D. Xu, Y. Xia, and D. P. Mandic, "Optimization in Quaternion Dynamic Systems: Gradient, Hessian, and Learning Algorithms," IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 2, pp. 249–261, Feb. 2016, issn: 2162-2388. doi: 10.1109/TNNLS.2015.2440473.

[25] G. Wahba, "A least squares estimate of satellite attitude," SIAM review, vol. 7, no. 3, pp. 409–409, 1965.

[26] F. L. Markley and J. L. Crassidis, Fundamentals of spacecraft attitude determination and control. Springer, 2014, vol. 33.

[27] T. R. Kane, P. W. Likins, and D. A. Levinson, Spacecraft Dynamics. McGraw Hill, 1983.

[28] J. Solà, "Quaternion kinematics for the error-state Kalman filter," en, arXiv:1711.02508 [cs], Nov. 2017. arXiv: 1711.02508 [cs]. [Online]. Available: http://arxiv.org/abs/1711.02508 (visited on 09/03/2019).

[29] T. Fan and T. Murphey, "Online Feedback Control for Input-Saturated Robotic Systems on Lie Groups," en, Robotics: Science and Systems XII, 2016. doi: 10.15607/RSS.2016.XII.027. arXiv: 1709.00376. [Online]. Available: http://arxiv.org/abs/1709.00376 (visited on 01/09/2020).

[30] G. Terzakis, M. Lourakis, and D. Ait-Boudaoud, "Modified Rodrigues Parameters: An Efficient Representation of Orientation in 3D Vision and Graphics," en, Journal of Mathematical Imaging and Vision, vol. 60, no. 3, pp. 422–442, Mar. 2018, issn: 0924-9907, 1573-7683. (visited on 12/19/2019).

[31] F. L. Markley and D. Mortari, "How to estimate attitude from vector observations," in Proceedings

of the AAS/AIAA Astrodynamics Specialist Conference, vol. 103, 1999, pp. 1979–1996.

[32] T. A. Howell, B. E. Jackson, and Z. Manchester, “Altro: A fast solver for constrained trajectory optimization,” in 2019 IEEE International Conference on Intelligent Robots and Systems, IEEE, 2019.

[33] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems.,” in ICINCO (1), 2004, pp. 222–229.

[34] Z. Manchester and S. Kuindersma, “Derivative-free trajectory optimization with unscented dynamic programming,” in 2016 IEEE 55th Conference on Decision and Control (CDC), IEEE, 2016, pp. 3642–3647.

[35] D. Mellinger, N. Michael, and V. Kumar, “Trajectory generation and control for precise aggressive maneuvers with quadrotors,” The International Journal of Robotics Research, vol. 31, no. 5, pp. 664–674, 2012.

[36] K. Tracy and Z. Manchester, “Model-predictive attitude control for flexible spacecraft during thruster firings,” in AAS/AIAA Astrodynamics Specialist Conference, Lake Tahoe, CA, Lake Tahoe, CA, Aug. 9, 2020.

[37] J. Brüdigam and Z. Manchester, “Linear-time variational integrators in maximal coordinates,” in Workshop on the Algorithmic Foundations of Robotics, Oulu, Finland, Jun. 2020.