Planning with Attitude

Brian 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 modifying the trajectory optimization solver ALTRO to plan directly on the space of unit quaternions, achieving superior convergence on problems involving significant changes in attitude.**

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 [29]. 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 [4] and $SO(3)$ [3]. Effective attitude tracking controllers have been developed for satellites [35], quadrotors [9, 20, 19, 13, 33, 24], and a 3D inverted pendulum [6] using various methods for calculating three-parameter attitude errors.

More recently, these ideas have been extended to trajectory generation [38], sample-based motion planning [39, 17], and optimal control. Approaches to optimal control with attitude states include analytical methods applied to satellites [28], discrete mechanics [16, 15, 18], a combination of sampling-based planning and constrained trajectory optimization for satellite formations [10, 2], projection operators [26], or more general theory for optimization on manifolds [34]. 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 [1, 7, 36, 11].

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 [22, 37], but significantly simpler and more general, enabling straight-forward adaptation of existing algorithms to systems with quaternion states. For concreteness, we then apply our method to the canonical Wahba's problem [32] 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. 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 using our notation.
- 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 $f(x) : \mathbb{R}^n \to \mathbb{R}$, $\frac{\partial f}{\partial x} \in \mathbb{R}^{1 \times n}$.

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 $\mathbf{q} \in \mathbb{H}$ as a standard vector $q \in \mathbb{R}^4 := [q_s \;\; q_v^T]^T$ 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 a double-cover of the rotation group $SO(3)$, since q and $-q$ represent the same rotation [23].

Quaternion multiplication is defined as

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

[1]Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, USA

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 $\mathbf{q}^{-1}$, giving the opposite rotation, is equal to its conjugate $\mathbf{q}^*$, which is simply the same quaternion with a negated vector part:

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

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 $\hat{r}' = \mathbf{q} \otimes \hat{r} \otimes \mathbf{q}^*$. 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 [14].

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:

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

where x and u are the state and control vectors, $r \in \mathbb{R}^3$ is the position, $\mathbf{q} \in \mathbb{S}^3$ 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.

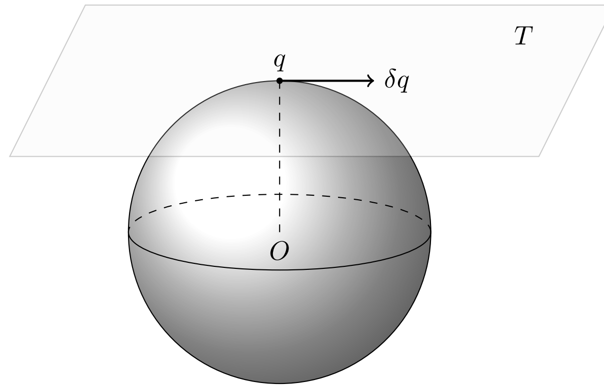

Fig. 1. When linearizing about a point q on an sphere $\mathbb{S}^{n-1}$ in n-dimensional space, the tangent space T is a plane living in $\mathbb{R}^{n-1}$, illustrated here with $n = 3$. Therefore, when linearizing about a unit quaternion $q \in \mathbb{S}^3$, the space of differential rotations lives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

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, 38, 18, 26, 27, 8, 34], while others have used the Cayley map (also known as Rodrigues parameters) [16, 15], Modified Rodrigues Parameters (MRPs) [30], or the vector part of the quaternion [9]. We choose Rodrigues parameters [23] 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

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

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:

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

The leading $G(q')^T$ comes from the fact that as $\phi' \to 0$, $L(q')f(q) \to I_q$, where I_q is the quaternion identity. 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. For constrained systems, the updated parameter can be projected back onto the feasible set at each iteration, but without the same convergence guarantees. For the constraints on $SO(3)$, Newton's method struggles to converge past a certain threshold due to this projection. By leveraging the quaternion calculus introduced, Newton's method can be modified to implicitly account for these constraints. To demonstrate this, we will examine Wahba's Problem. In 1965, Grace Wahba proposed the criterion for a least squares estimate of a spacecraft's attitude from vector measurements [32, 23]. We will solve this problem using a modified version of Newton's method that exploits the true group structure of $SO(3)$ using the quaternion calculus presented here.

A. Methodology

Given known vectors in some world frame, ${}^W w_i$, and measurements of these vectors in some body fixed frame, ${}^B v_i$, our goal is to determine the relative rotation from the body to world frame ${}^W A(q)^B$, expressed as a quaternion. We can define Wahba's loss function as the following:

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

where $r_i(q)$ is the residual vector.

We can solve for ${}^W q^B$ using a nonlinear least squares method minimizing Wahba's loss function:

$$\begin{aligned} \underset{q}{\text{minimize}} \quad & \|r(q)\|_2^2 \\ \text{subject to} \quad & q \in \mathbb{S}^3. \end{aligned}$$

Following the typical approach for Newton's method, we minimize (19) by setting the gradient to zero:

$$\begin{aligned} \frac{\partial L}{\partial q}^T &= \sum_i \frac{\partial r(q)}{\partial q}^T r_i(q) := \bar{J}^T r(q) = 0 \\ &= \sum_i (-2H^T R(q)^T R({}^B\hat{v}_i)^T r(q). \end{aligned} \tag{20}$$

which can be obtained from the chain rule and (9).

Applying the methodology from the previous section, we actually want to minimize (19) with respect to a differential rotation ϕ, which we do by simply "correcting" our Jacobian $\bar{J}$ using (18):

$$J = \frac{\partial r(\mathbf{q} \otimes \varphi(\phi))}{\partial \phi} = \frac{\partial r(q)}{\partial q} G(q). \tag{21}$$

We then use this to obtain our Newton step by using the Moore-Penrose psuedoinverse: $\phi = (J^T J)^{-1} J^T$. To obtain our next iterate, we "add" the step using the correct notion of composition for the group: $\mathbf{q}_{k+1} = \mathbf{q}_k \otimes \varphi(\phi)$. This "multiplicative" Newton algorithm is summarized in Algorithm 1.

Algorithm 1 Multiplicative Gauss-Newton Method

1: $k = 0$
2: **while** significant progress **do**
3: $\quad J = \frac{\partial r(q_k)}{\partial q} G(q_k)$ ▷ quaternion adjusted Jacobian
4: $\quad \phi = -(J^T J)^{-1} J^T r(q_k)$
5: $\quad \mathbf{q}_{k+1} = \mathbf{q}_k \otimes \varphi(\phi)$ ▷ apply step multiplicatively
6: $\quad k = k + 1$
7: **end while**

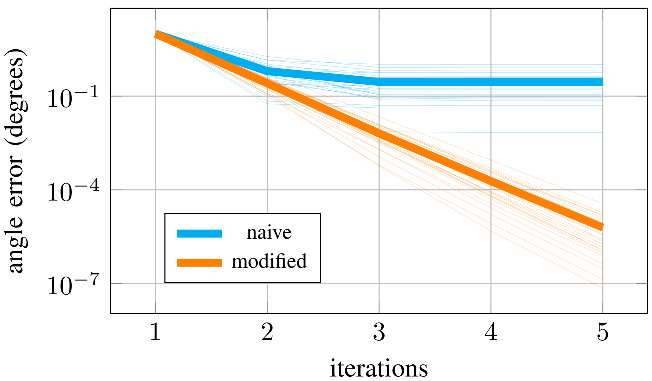

Fig. 2. Convergence comparison for Wahba's problem. By performing Newton's method on the error quaternion and applying the result to the full quaternion we achieve quadratic convergence, whereas the more naïve approach doesn't converge to zero error. The angle error is calculated relative to the true analytical solution obtained via an SVD decomposition.

B. Results

We compared this modified Newton's method with a naïve Newton's method where the quaternion is simply re-normalized at every iteration. As illustrated in Figure 2, it is clear that the naive original method makes progress initially, but fails to exhibit the quadratic convergence typical of a Newton method. By optimizing directly in the space of unit quaternions, the modified method is able to achieve the expected quadratic convergence. It should also be clear from the previous section that the adaptations to the original Newton method are simple and straightforward, highlighting both the effectiveness and value of the proposed approach.

V. Trajectory Optimization on $\mathbb{R}^n \times SO(3)$

Here we outline the modifications to the ALTRO solver [12], to solve trajectory optimization problems for rigid bodies, which extends easily to arbitrary systems whose state is in $\mathbb{R}^n \times SO(3)$.

We consider trajectory optimization problems of the form,

$$\begin{aligned} \underset{x_{0:N}, u_{0:N-1}}{\text{minimize}} \quad & \ell_f(x_N) + \sum_{k=0}^{N-1} \ell_k(x_k, u_k) \\ \text{subject to} \quad & x_{k+1} = f(x_k, u_k), \\ & g_k(x_k, u_k) \leq 0, \\ & h_k(x_k, u_k) = 0, \end{aligned} \tag{22}$$

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.

Like most gradient or Newton-based methods for optimization, ALTRO approximates the nonlinear functions $f, \ell, g,$ 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 $\delta x \in \mathbb{R}^{12}$.

We begin by linearizing the dynamics about the reference trajectory using (18). Our linearized error dynamics become

$$\delta x_{k+1} = A_k \delta x_k + B_k \delta u_k \tag{23}$$

where

$$\begin{aligned} A_k &= E(\bar{x}_{k+1})^T \frac{\partial f}{\partial x}|_{\bar{x}_k, \bar{u}_k} E(\bar{x}_k), \\ B_k &= E(\bar{x}_{k+1})^T \frac{\partial f}{\partial u}|_{\bar{x}_k, \bar{u}_k}, \end{aligned} \tag{24}$$

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

$$\delta x_k = \begin{bmatrix} r_k - \bar{r}_k \\ \varphi^{-1}(\bar{\mathbf{q}}_k^{-1} \otimes \mathbf{q}_k) \\ v_k - \bar{v}_k \\ \omega_k - \bar{\omega}_k \end{bmatrix}, \; E(x) = \begin{bmatrix} I_3 & & & \\ & G(q) & & \\ & & I_3 & \\ & & & I_3 \end{bmatrix}. \tag{25}$$

By applying (14) and (15) to our 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:

$$\begin{aligned} \delta\ell(x, u) \approx \frac{1}{2} \delta x^T \ell_{xx} \delta x + \frac{1}{2} \delta u^T \ell_{uu} \delta u + \delta_u^T \ell_{ux} \delta u \\ + \ell_x^T \delta x^T + \ell_u^T \delta u. \end{aligned} \tag{26}$$

With this expansion, we calculate the expansion of the "action-value function" $Q(x, u)$ as normal, including the extra terms from the augmented Lagrangian cost:

$$Q_{xx} = \ell_{xx} + A_k^T P_{k+1} A_k \tag{27}$$
$$Q_{uu} = \ell_{uu} + B_k^T P_{k+1} B_k \tag{28}$$
$$Q_{ux} = \ell_{ux} + B_k^T P_{k+1} A_k \tag{29}$$
$$Q_x = \ell_x + A_k^T p_{k+1} \tag{30}$$
$$Q_u = \ell_u + B_k^T p_{k+1}, \tag{31}$$

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 linearized feedback gains $K_k \in \mathbb{R}^{m \times 12}$ and $d_k \in \mathbb{R}^m$ by starting at the terminal state and resursing backward in time along the trajectory during the "backward pass" of the iLQR algorithm. During the "forward pass", the dynamics are simulated forward in time using the feedback gains computed during the backward pass. At each time step, the control is calculated using the linear feedback controller:

$$u_k = K_k \delta x_k + \bar{u}_k. \tag{32}$$

where $\bar{u}_k$ is the control value from the previous iteration, and δx is computed using (25), with x_k being the current state estimate and $\bar{x}_k$ the state from the previous iteration. The rest of the algorithm is left unchanged. For more details on the ALTRO algorithm, the reader is encouraged to refer to the original paper [12].

Problem	Iterations	time (ms)
barrellroll	53 / 40	94.93 / 76.48
quadflip	58 / 22	419.08 / 138.63
satellite	35 / 35	392.66 / 460.08

TABLE I
TRAJECTORY OPTIMIZATION TIMING RESULTS (NAIVE/MODIFIED)

A. Quaternion Cost Functions

In addition to the straight-forward modifications to the ALTRO algorithm itself, we need to carefully consider the types of cost functions we minimize. 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 is ill-defined. We propose the following cost function which penalizes the geodesic distance between two unit quaternions [17], which we have found to work well in practice. For sake of clarity and space, we only consider the costs on the quaternion variables: costs on the other states and the control variables remain unaffected.

$$J_{\text{geo}} = (1 - |q_g^T q|), \tag{33}$$

whose gradient and Hessian are,

$$\nabla J_{\text{geo}} = \text{sign}(q_d^T q) q_g^T G(q) \tag{34}$$
$$\nabla^2 J_{\text{geo}} = \text{sign}(q_d^T q) I_3 q_g^T q, \tag{35}$$

where sign denotes the signum function.

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 results were run on a desktop computer with an AMD Ryzen 2950x processor with 40 GB of RAM. All problems are run using ALTRO, first without any of the modifications presented in the current paper, analagous to the naive 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". Timing results are summarized in Table VI. All experiments were solved to a constraint satisfaction tolerance of 10^{-5} and discretized with a 4th order Runge-Kutta integrator. Code for all experiments is available on GitHub[1].

A. Satellite Attitude Keep-Out

A spacecraft with flexible appendages must perform a 150 degree slew while ensuring that a body-mounted camera does not cross within 40 degrees of the sun. The attitudes that result in the camera pointing too close to the sun will be referred to as a keep-out zone. The flexible body spacecraft dynamics are based on equation (10), with the addition of four reaction wheels, and the three largest flexible modes [31]. In order to ensure no damage to the camera, we must formulate a constraint that guarantees that the camera line-of-sight body-fixed vector (${}^B r_{cam}$) does not look within 40 degrees of the world-fixed sun vector (${}^W r_{sun}$). This angle constraint can be represented as a dot product between the two vectors expressed in the world frame:

$$\left({}^W r_{sun}\right)^T \left({}^W A(q)^B \, {}^B r_{cam}\right) \le \cos(40°). \tag{36}$$

The attitudes that satisfy this constraint comprise a non-convex set, with the constraint itself being nonlinear in the attitude quaternion.

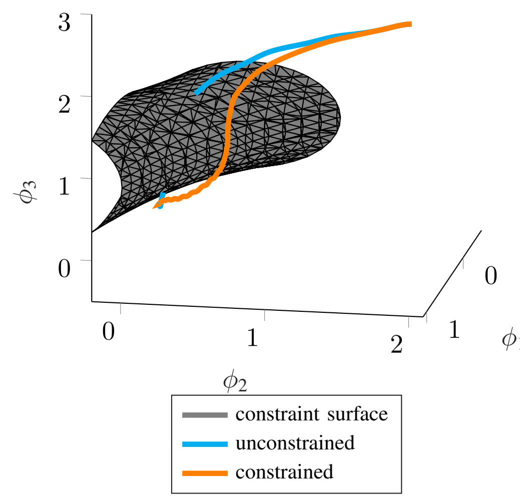

Fig. 3. Visualization of the flexible spacecraft slew with a keep-out zone. Attitude is parameterized with a Rodrigues parameter to visualize the trajectory in three dimensions. The constraint surface represents attitudes where the camera line-of-sight is within 40° of the sun. The unconstrained solution violates this constraint, while the constrained solution is able to avoid the keep out zone.

ALTRO is able to reason about this nonlinear constraint, and account for the group structure present in the quaternion. From Figure 3 we can see that without the camera constraint present, the trajectory will pass through the keep-out zone, making the trajectory infeasible. With the keep-out zone constraint present, ALTRO is able to converge on a trajectory that performs the slew in an efficient manner, while keeping the camera sufficiently far from the sun.

The modifications didn't have a significant impact on runtime performance for this example. In our experience, the computational benefits of the proposed methodology are problem-dependent, especially for highly nonlinear systems. The following examples result in more dynamic and aerobatic behaviors, where the computational benefit of the proposed methodology is significant, including an example where the naive method failed to find a good solution.

B. Airplane Barrel Roll

Our airplane model is a simple rigid body as defined in Section II-B with forces and torques due to lift and drag

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

fit from wind tunnel data [21]. The airplane was tasked to do a barrell roll by setting a high terminal cost for being upside-down, see Fig. 4. The solver is initialized with level flight trim conditions. The convergence of the different versions of ALTRO is compared in Fig. 5. 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.

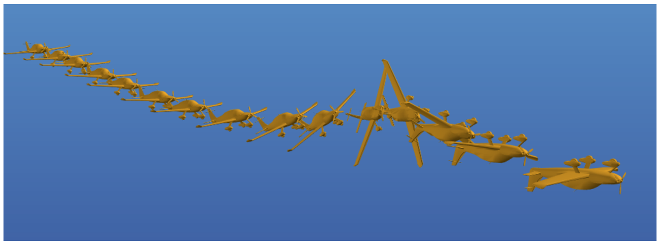

Fig. 4. Barrel roll trajectory computed by iterative MLQR using a terminal cost to encourage an upside-down attitude.

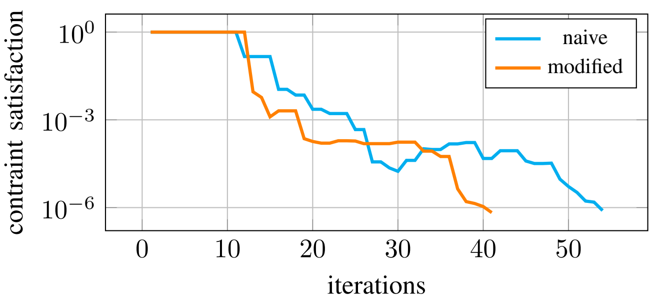

Fig. 5. Constraint convergence when solving the barrel roll problem. Compares the convergence of the original version of ALTRO versus the new, modified version that optimizing on the error state.

C. Quadrotor Flip

We optimized a 360 degree flip trajectory for a quadrotor with dynamics adapated from [25] using the modified version of ALTRO. Four intermediary "keyframes" were used to encourage the quadrotor to be at angles of 90°, 180°, 270°, and 360° around an approximately circular arc. The quadrotor was constrained to stay above the floor and move to a goal state 2 meters away in the $+y$ direction. The solver was initialized with a dynamically infeasible trajectory that linearly interpolates between the initial and final states, rotating the quad around the x-axis a full 360°.

Figure 6 shows snapshots of the trajectory as generated using ALTRO. The original version of ALTRO, even after significant tuning efforts, was not able to converge to the desired solution. This behavior is common and expected when attempting optimization that does not properly account for the group structure of rotations. 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

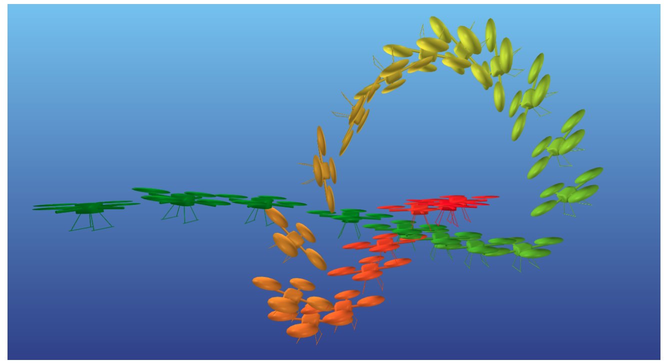

Fig. 6. Snapshots of the quadrotor flip trajectory. The gree-colored quadrotors represent the state near t=0 s and the red-colored quadrotors represent the state near t=5.0 s

Euler angles, Rodrigues parameters, and Modified Rodrigues Parameters, respectively.

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. We have demonstrated that the application of this methodology is straightforward and yields substantial improvements in the convergence of Newton-based methods (see Fig. 2), while also offering improvements for nonlinear constrained trajectory optimization for floating-base systems (see Table VI).

With the modifications presented, ALTRO can solve problems few other methods for trajectory optimization can. Many state-of-the-art methods, such as direct collocation or sequential convex programming, rely on commercial, proprietary, or general-purpose solvers whose internal numerics often are abstracted away from the user. By exploiting the unique structure of both the trajectory optimization problem and the rotation group, ALTRO will likely be able to solve more challenging problems with performance.

The methods presented here can easily be leveraged to adapt other classes of gradient or Newton-based algorithms to exploit the structure of 3D rotations. Future work may include adaptation of methods for state estimation, localization, design, or other methods for motion planning such as direct collocation.

Future work will focus on continued refinement of ALTRO and performance improvements for use in real-time model-predictive control applications, as well as extensions to multi-body robotic systems, such as humanoids or quadrupeds, represented in "maximal" coordinates that include the 3D orientation of each body [5].

Acknowledgements: 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

TODO: Add NASA acknowledgements (JPL + ECF)

References

[1] Yaser Alothman and Dongbing Gu. "Quadrotor Transporting Cable-Suspended Load Using Iterative Linear Quadratic Regulator (iLQR) Optimal Control". In: *2016 8th Computer Science and Electronic Engineering (CEEC)*. Sept. 2016, pp. 168–173. DOI: 10.1109/CEEC.2016.7835908.

[2] Georges S. Aoude, Jonathan P. How, and Ian M. Garcia. "Two-Stage Path Planning Approach for Solving Multiple Spacecraft Reconfiguration Maneuvers". en. In: *The Journal of the Astronautical Sciences* 56.4 (Dec. 2008), pp. 515–544. ISSN: 0021-9142. DOI: 10.1007/BF03256564. URL: http://link.springer.com/10.1007/BF03256564 (visited on 01/08/2020).

[3] J. Baillieul. "Geometric Methods for Nonlinear Optimal Control Problems". en. In: *Journal of Optimization Theory and Applications* 25.4 (Aug. 1978), pp. 519–548. ISSN: 0022-3239, 1573-2878. DOI: 10.1007/BF00933518. URL: http://link.springer.com/10.1007/BF00933518 (visited on 01/07/2020).

[4] R. W. Brockett. "Lie Theory and Control Systems Defined on Spheres". en. In: *SIAM Journal on Applied Mathematics* 25.2 (Sept. 1973), pp. 213–225. ISSN: 0036-1399, 1095-712X. DOI: 10.1137/0125025. URL: http://epubs.siam.org/doi/10.1137/0125025 (visited on 01/07/2020).

[5] Jan Brüdigam and Zachary Manchester. "Linear-Time Variational Integrators in Maximal Coordinates". In: *arXiv* (2020).

[6] Nalin A. Chaturvedi, N. Harris McClamroch, and Dennis S. Bernstein. "Asymptotic Smooth Stabilization of the Inverted 3-D Pendulum". In: *IEEE Transactions on Automatic Control* 54.6 (June 2009), pp. 1204–1215. ISSN: 2334-3303. DOI: 10.1109/TAC.2009.2019792.

[7] Cedric de Crousaz et al. "Unified Motion Control for Dynamic Quadrotor Maneuvers Demonstrated on Slung Load and Rotor Failure Tasks". en. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA: IEEE, May 2015, pp. 2223–2229. ISBN: 978-1-4799-6923-4. DOI: 10.1109/ICRA.2015.7139493. URL: http://ieeexplore.ieee.org/document/7139493/ (visited on 01/09/2020).

[8] Taosha Fan and Todd Murphey. "Online Feedback Control for Input-Saturated Robotic Systems on Lie Groups". en. In: *Robotics: Science and Systems XII* (2016). DOI: 10.15607/RSS.2016.XII.027. arXiv: 1709.00376. URL: http://arxiv.org/abs/1709.00376 (visited on 01/09/2020).

[9] Emil Fresk and George Nikolakopoulos. "Full Quaternion Based Attitude Control for a Quadrotor". In: *2013 European Control Conference (ECC)*. July 2013, pp. 3864–3869. DOI: 10.23919/ECC.2013.6669617.

[10] I. Garcia and J.P. How. "Trajectory Optimization for Satellite Reconfiguration Maneuvers with Position and Attitude Constraints". In: *Proceedings of the 2005, American Control Conference, 2005.* June 2005, 889–894 vol. 2. DOI: 10.1109/ACC.2005.1470072.

[11] Mathieu Geisert and Nicolas Mansard. "Trajectory Generation for Quadrotor Based Systems Using Numerical Optimal Control". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. May 2016, pp. 2958–2964. DOI: 10.1109/ICRA.2016.7487460.

[12] Taylor A Howell, Brian E Jackson, and Zachary Manchester. "ALTRO: A fast solver for constrained trajectory optimization". In: *2019 IEEE International Conference on Intelligent Robots and Systems, IEEE.* 2019.

[13] Eric N. Johnson and Suresh K. Kannan. "Adaptive Trajectory Control for Autonomous Helicopters". en. In: *Journal of Guidance, Control, and Dynamics* 28.3 (May 2005), pp. 524–538. ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.6271. URL: https://arc.aiaa.org/doi/10.2514/1.6271 (visited on 01/07/2020).

[14] Thomas R Kane, Peter W Likins, and David A Levinson. *Spacecraft Dynamics*. McGraw Hill, 1983.

[15] Marin 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.

[16] Marin B. Kobilarov and Jerrold E. Marsden. "Discrete Geometric Optimal Control on Lie Groups". In: *IEEE Transactions on Robotics* 27.4 (Aug. 2011), pp. 641–655. ISSN: 1941-0468. DOI: 10.1109/TRO.2011.2139130.

[17] J.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. URL: http://ieeexplore.ieee.org/document/1308895/ (visited on 01/08/2020).

[18] T. Lee, M. Leok, and N. H. McClamroch. "Optimal Attitude Control of a Rigid Body Using Geometrically Exact Computations on SO(3)". en. In: *Journal of Dynamical and Control Systems* 14.4 (Oct. 2008), pp. 465–487. ISSN: 1079-2724, 1573-8698. DOI: 10.1007/s10883-008-9047-7. URL: http://link.springer.com/10.1007/s10883-008-9047-7 (visited on 12/19/2019).

[19] Taeyoung Lee, Melvin Leok, and N Harris 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.

[20] Hao Liu, Xiafu Wang, and Yisheng Zhong. "Quaternion-Based Robust Attitude Control for Uncertain Robotic Quadrotors". In: *IEEE Transactions on Industrial Informatics* 11.2 (Apr. 2015), pp. 406–415. ISSN: 1941-0050. DOI: 10.1109/TII.2015.2397878.

[21] Zachary Manchester and Scott Kuindersma. "Derivative-Free Trajectory Optimization with Unscented Dynamic Programming". In: *IEEE Conference on Decision and Control (CDC)*. pubs-conference. Las Vegas, NV: IEEE, Dec. 2016. URL: https://rexlab.stanford.edu/papers/udp.pdf.

[22] D. P. Mandic, C. Jahanchahi, and C. Cheong Took. "A Quaternion Gradient Operator and Its Applications". In: *IEEE Signal Processing Letters* 18.1 (Jan. 2011), pp. 47–50. ISSN: 1558-2361. DOI: 10.1109/LSP.2010.2091126.

[23] F Landis Markley and John L Crassidis. *Fundamentals of spacecraft attitude determination and control*. Vol. 33. Springer, 2014.

[24] Daniel Mellinger and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors". In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 2520–2525.

[25] Daniel Mellinger, Nathan Michael, and Vijay Kumar. "Trajectory generation and control for precise aggressive maneuvers with quadrotors". In: *The International Journal of Robotics Research* 31.5 (2012), pp. 664–674.

[26] Alessandro Saccon, John Hauser, and A. Pedro Aguiar. "Optimal Control on Lie Groups: The Projection Operator Approach". In: *IEEE Transactions on Automatic Control* 58.9 (Sept. 2013), pp. 2230–2245. ISSN: 2334-3303. DOI: 10.1109/TAC.2013.2258817.

[27] Joan Solà. "Quaternion Kinematics for the Error-State Kalman Filter". en. In: *arXiv:1711.02508 [cs]* (Nov. 2017). arXiv: 1711.02508 [cs]. URL: http://arxiv.org/abs/1711.02508 (visited on 09/03/2019).

[28] Karlheinz Spindler. "Optimal Control on Lie Groups with Applications to Attitude Control". en. In: *Mathematics of Control, Signals, and Systems* 11.3 (Sept. 1998), pp. 197–219. ISSN: 0932-4194, 1435-568X. DOI: 10.1007/BF02741891. URL: http://link.springer.com/10.1007/BF02741891 (visited on 01/07/2020).

[29] John Stuelpnagel. "On the parametrization of the three-dimensional rotation group". In: *SIAM review* 6.4 (1964), pp. 422–430.

[30] George Terzakis, Manolis Lourakis, and Djamel Ait-Boudaoud. "Modified Rodrigues Parameters: An Efficient Representation of Orientation in 3D Vision and Graphics". en. In: *Journal of Mathematical Imaging and Vision* 60.3 (Mar. 2018), pp. 422–442. ISSN: 0924-9907, 1573-7683. (Visited on 12/19/2019).

[31] Kevin Tracy and Zachary Manchester. "Model-Predictive Attitude Control for Flexible Spacecraft During Thruster Firings". In: *AAS/AIAA Astrodynamics Specialist Conference*. Lake Tahoe, CA, Aug. 9, 2020.

[32] G. Wahba. "A Least Squares Estimate of Satellite Attitude". In: *SIAM Review* 7.3 (July 1, 1965), pp. 409–409. ISSN: 0036-1445. DOI: 10.1137/1007077. URL: http://epubs.siam.org/doi/abs/10.1137/1007077.

[33] Michael Watterson and Vijay Kumar. "Control of quadrotors using the HOPF fibration on SO (3)". In: *Robotics Research*. Springer, 2020, pp. 199–215.

[34] Michael Watterson et al. "Trajectory Optimization On Manifolds with Applications to $SO(3)$ and $\mathbb{R}^3 \times S^2$." In: *Robotics: Science and Systems*. 2018.

[35] Bong Wie and Peter M Barba. "Quaternion feedback for spacecraft large angle maneuvers". In: *Journal of Guidance, Control, and Dynamics* 8.3 (1985), pp. 360–365.

[36] Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. "Model Predictive Path Integral Control: From Theory to Parallel Computation". en. In: *Journal of Guidance, Control, and Dynamics* 40.2 (Feb. 2017), pp. 344–357. ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.G001921. URL: http://arc.aiaa.org/doi/10.2514/1.G001921 (visited on 01/09/2020).

[37] Dongpo Xu, Yili Xia, and Danilo P. Mandic. "Optimization in Quaternion Dynamic Systems: Gradient, Hessian, and Learning Algorithms". In: *IEEE Transactions on Neural Networks and Learning Systems* 27.2 (Feb. 2016), pp. 249–261. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2015.2440473.

[38] M. Zefran, V. Kumar, and C.B. Croke. "On the Generation of Smooth Three-Dimensional Rigid Body Motions". In: *IEEE Transactions on Robotics and Automation* 14.4 (Aug. 1998), pp. 576–589. ISSN: 2374-958X. DOI: 10.1109/70.704225.

[39] Miloš Žefran, Vijay Kumar, and Christopher Croke. "Metrics and Connections for Rigid-Body Kinematics". en. In: *The International Journal of Robotics Research* 18.2 (Feb. 1999), pp. 242–1. ISSN: 0278-3649.