Planning with Attitude

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

***Abstract*— Planning and controlling 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 based on vector calculus and linear algebra. We demonstrate the effectiveness of the approach on a simple application of Newton's method to the canonical Wabha problem and achieve quadratic convergence while optimizing directly on the space of $SO(3)$. We also apply the methodology the modify the ALTRO algorithm for constrained nonlinear trajectory optimization to correctly account for the group structure of rotations during the course of the optimization. Comparisons on a few trajectory optimization problems for floating-base systems undergoing significant changes in attitude highlight the effectiveness of the proposed approach. *Index Terms*— otion planning and control, quaternions, optimal control, linear quadratic regulator otion planning and control, quaternions, optimal control, linear quadratic regulator m**

I. INTRODUCTION

Many useful 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 [27]. As a result, when parameterizing rotations, we must either a) pick a three-parameter representation and deal with discontinuities, or b) pick 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 [31], quadrotors [9, 20, 19, 13, 29, 23], 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 [34], sample-based motion planning [35, 17], and optimal control. Approaches to optimal control on attitude problems include analytical methods applied to satellites [26], discrete mechanics [16, 15, 18], a combination of sampling-based planning and constrained trajectory optimization for satellite formations [10, 2], projection operators [24], or more general theory for optimization on manifolds [30]. 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 apply traditional methods for motion planning and control with no regard for the group structure of rigid body motion [1, 7, 32, 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 calculus that should be familiar to most roboticists. Similar to [21, 33], in Sec. III we introduce a quaternion differential calculus, but take a significantly simpler and more general approach, enabling straight-forward adaptation of existing algorithms to systems with quaternion states. To make this concrete, in Sec. IV we apply this method to the canonical Wahba's problem and demonstrate dramatically 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 the performance gains on several constrained benchmark problems. To our knowledge, there does not currently exist a solver that is capable of leveraging the unique Markovian structure of the fixed-horizon trajectory optimization problem while correctly accounting for the group structure of 3D rotations. In summary, our contributions include:

- A unified approach to quaternion differential calculus entirely based on standard vector calculus and linear algebraic operations control of systems with quaternion states
- 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 during the solve

II. BACKGROUND

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

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

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 the scalar and vector part of the quaternion, respectively.

Quaternion multiplication is defined as

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

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:

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

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

In the current work we will restrict our focus to rigid bodies moving freely in 3D space. That is, we consider systems with dynamics of the following form:

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

where x and u are the state and control vectors, $r \in \mathbb{R}^3$ is the position, $R \in SO(3)$ 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, $J \in \mathbb{R}^{3\times 3}$ is the inertia matrix, $F_G(x,u) \in \mathbb{R}^3$ are the forces in the global frame, and $\tau_L(x,u)$ are the moments in the local (body) frame.

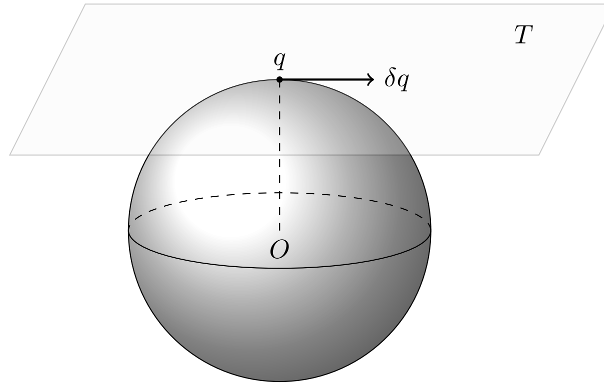

Fig. 1. When linearizing about a point q on an sphere $\mathbb{S}^{n-1}$ in n-dimensional space, the tangent space T is a plane living in $\mathbb{R}^{n-1}$, illustrated here with $n = 3$. Therefore, when linearizing about a unit quaternion $q \in \mathbb{S}^3$, the space of differential rotations lives 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 velocity, 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, 34, 18, 24, 25, 8, 30], while others have used the Cayley map (also known as Rodrigues parameters) [16, 15], Modified Rodrigues Parameters (MRPs) [28], or the vector part of the quaternion [9]. We choose Rodrigues parameters [22] 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 $\phi = 0$:

$$\nabla h(q) = \frac{\partial h}{\partial q} L(q) H := \frac{\partial h}{\partial q} G(q) = \frac{\partial h}{\partial q} \begin{bmatrix} -q_v^T \\ sI_3 + [q_v]^\times \end{bmatrix} \tag{14}$$

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 difference or automatic differentiation. As an aside, although we have used Rodrigues parameters, $G(q)$ is actually the same (up to a constant scaling 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 ends up being the same (up to a scaling 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 value 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. Differentiating through the inverse map, evaluated at the quaternion identity, we find that $\partial\varphi^{-1} / \partial q \to H^T$ 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 [22]. We will solve this problem using a standard nonlinear least squares method, as well as a method that exploits the true group structure of $SO(3)$ using the quaternion calculus presented here.

A. Methodology

Given known vectors in some inertial frame, ${}^N v_i$, 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 inertial frames ${}^N q^N$, expressed as a quaternion. We can define Wahba's loss function as the following:

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

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

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

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

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

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

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

Treating q as a vector in $\mathbb{R}^4$, we obtain a solution to (20) using the Moore-Penrose pseudoinverse, $\delta q = (J^T J)^{-1} J^T$, and our next candidate quaternion via simple addition, $q_{k+1} = q_k + \delta q$. Since q_{k+1} will no longer be unit norm, we project it back on the unit sphere via the projection operator $\Pi(q) = q/\|q\|$. This "projected" Newton approach is summarized in Algorithm 1.

Alternatively, if we instead minimize with respect to a differential quaternion ϕ, we adapt the algorithm by simply "correcting" our Jacobian J using (18):

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

Algorithm 1 Projected Gauss-Newton Method

$k = 0$
while significant progress **do**
  $J = \frac{\partial r(q_k)}{\partial \partial q}$
  $\delta q = -(J^T J)^{-1} J^T r(q_k)$
  $q_{k+1} = \Pi_{SO(3)}(q_k + \delta q)$
  $k = k + 1$
end while

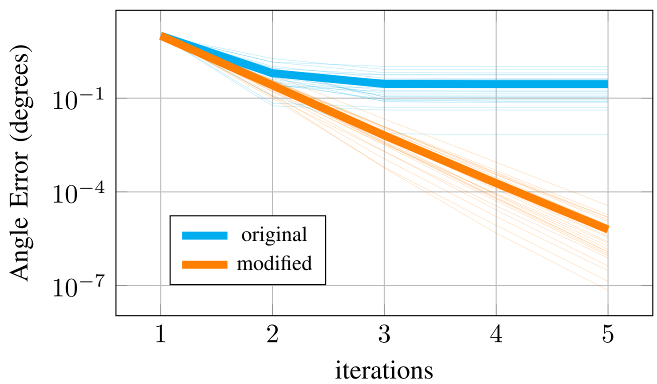

Fig. 2. Convergence comparison for Wahba's problem. By performing Newton's method on the error quaternion and applying the result to the full quaternion we achieve quadratic convergence, whereas the more naïve approach doesn't converge to zero error. The angle error is calculated relative to the true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

We obtain our step—this time in the actual tangent space—as before: $\phi = (\bar{J}^T \bar{J})^{-1} \bar{J}^T)$. To obtain our next iterate, we "add" the step using the correct notion of composition for the group: $q_{k+1} = q_k \otimes \varphi\phi$. This "multiplicative" Newton algorithm is summarized in Algorithm 2.

Algorithm 2 Multiplicative Gauss-Newton Method

$k = 0$
while significant progress **do**
  $\bar{J} = \frac{\partial r(q_k)}{\partial q} G(q_k)$ ▷ quaternion adjusted Jacobian
  $\phi = -(\bar{J}^T \bar{J})^{-1} \bar{J}^T r(q_k)$
  $q_{k+1} = q_n \otimes \varphi\phi$ ▷ apply step multiplicatively
  $k = k + 1$
end while

B. Results

As illustrated in Figure 2, it is clear that by projecting back onto the unit quaternion at each iteration make initial progress, but fails to exhibit the quadratic convergence typical of a Newton method. By optimizing directly in the space of differential quaternions, we 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 in $R^n \times SO(3)$. ALTRO is an efficient solver for constrained nonlinear optimization problems that uses iterative LQR (iLQR) with an augmented Lagrangian framework.

We consider a trajectory optimization problem of the form:

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

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 is a general nonlinear cost function, 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, 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 $\delta x \in \mathbb{R}^{12}$.

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

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

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:

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

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

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

of the augmented Lagrangian cost function:

$$\mathcal{L}_A = \mathcal{L}_N(x_N, \lambda_N, \mu_N) + \sum_{k=0}^{N-1} \mathcal{L}_k(x_k, u_k, \lambda_k, \mu_k) \tag{27}$$

where

$$\mathcal{L}_k(x, u, \lambda, \mu) = \ell(x,u) + (\lambda + \frac{1}{2} I_\mu c(x,u))^T c(x,u), \tag{28}$$

with $c(x,u)$ being the concatenation of the constraints f, g, and h at a given time step, and I_μ the penalty matrix.

With this expansion, we calculate the expansion of the "action-value function" $Q(x, u)$ as normal:

$$Q_{xx} = \ell_{xx} + A_k^T P_{k+1} A_k \tag{29}$$
$$Q_{uu} = \ell_{uu} + B_k^T P_{k+1} B_k \tag{30}$$
$$Q_{ux} = \ell_{ux} + B_k^T P_{k+1} A_k \tag{31}$$
$$Q_x = \ell_x + A_k^T p_{k+1} \tag{32}$$
$$Q_u = \ell_u + B_k^T p_{k+1}, \tag{33}$$

from which we can calculate the quadratic expansion of the cost-to-go $P_k \in \mathbb{R}^{12\times 12}$, $p_k \in \mathbb{R}^1 2$, 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:

$$u_k = K_k \delta x_k + \bar{u}_k. \tag{34}$$

where $\bar{u}_k$ is the control value from the previous iteration, and δx is computed using (25), with x_k 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].

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 two different cost functions that accomplish similar behavior. 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.

1) Error Quadratic: Rather than simple subtraction, we can use a quadratic function on the three-parameter error state (25):

$$J_{\text{err}} = \frac{1}{2}\phi^T Q \phi = \frac{1}{2}\left(\varphi^{-1}(\delta q)\right)^T Q \left(\varphi^{-1}(\delta q)\right). \tag{35}$$

where $\delta q = L(q_g)^T q$, and $\phi = \varphi \delta q$. The gradient and Hessian of (35) are

$$\nabla J_{\text{err}} = \phi^T Q D(\delta q) G(\delta q) \tag{36}$$

$$\nabla^2 J_{\text{err}} = G(\delta q)^T \left(D(\delta q)^T Q D(\delta q) + \nabla D\right) G(\delta q) + I_3(\phi^T Q D(\delta q))\delta q \tag{37}$$

where, for the Cayley map,

$$D(q) = \frac{\partial \varphi^{-1}}{\partial q} = -\frac{1}{q_s^2}\begin{bmatrix} q_v & -\frac{1}{q_s} I_3 \end{bmatrix} \tag{38}$$

$$\nabla D = \frac{\partial}{\partial q}\left(D(q)^T Q \phi\right) = -\frac{1}{q_s^2}\begin{bmatrix} -2\frac{q_v}{q_s}^T Q\phi & \phi^T Q \\ Q\phi & 0 \end{bmatrix}. \tag{39}$$

Problem	Iterations	time (ms)
barrellroll	47 / 36	94.54 / 78.65
quadflip	58 / 28	457.58 / 217.69
satellite	35 / 35	446.52 / 517.47

TABLE I
TRAJECTORY OPTIMIZATION TIMING RESULTS

2) Geodesic Distance: Alternatively, we can use the geodesic distance between two quaternions [17],

$$J_{\text{geo}} = (1 - \left|q_g^T q\right|), \tag{40}$$

whose gradient and Hessian are,

$$\nabla J_{\text{geo}} = \pm q_g^T G(q) \tag{41}$$
$$\nabla^2 J_{\text{geo}} = \pm I_3 q_g^T q, \tag{42}$$

where the sign of the Hessian corresponds to the sign of $q_d^T q$.

VI. EXPERIMENTS

In this section we present several trajectory optimization problems for systems that undergo large changes in attitude: an airplane barrell roll, a quadrotor flip, and a satellite with flexible modes that must slew to a new orientation while avoiding a "keep out" orientation 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 modification presented in the current paper, labeled "original", and then using the modifications listed in Sec. V and the geodesic cost funtion described in Sec. V-A.2. Timing results are summarized in Table VI. All experiments were solved to a constraint satisfaction tolerance of 10^{-5}. Code for all experiments is available on GitHub[1].

A. Satellite Attitude Keep-Out

B. Airplane Barrel Roll

An airplane model with aerodynamic coefficients fit from real wind-tunnel data is tasked to do a barrel 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 is compared in Fig. 5. For both the quadrotor flip and the airplane barrel roll, the modified version of ALTRO converged faster than original version. For these highly aerobatic maneuvers, we achieve, as expected, better performance by correctly leveraging the structure of the rotation group during the optimization routine.

C. Quadrotor Flip

We sucessfully optimized a 360 degree flip trajectory for a quadrotor using the modified version of ALTRO. Four intermediary knotpoints 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

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

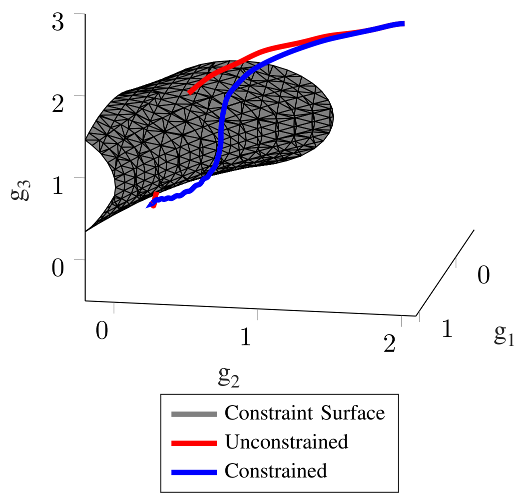

Fig. 3. Attitude keep out

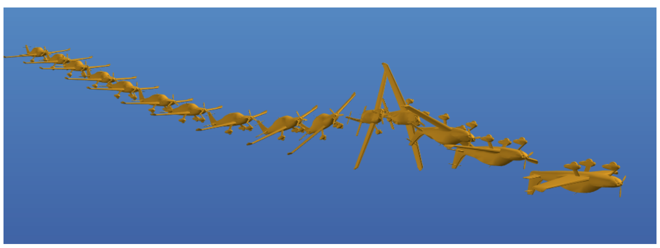

Fig. 4. Barrel roll trajectory computed by iterative MLQR using a terminal cost to encourage an upside-down attitude.

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, would not converge to the desired solution. It instead got halfway and then "unwound" 360 degrees and then continued rotating to the final orientation. This behavior is common and expected when attempting optimization that does not properly account for the grou structure of rotations.

VII. Conclusions

We have presented a general, unified method for planning and control on 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 substantion 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.

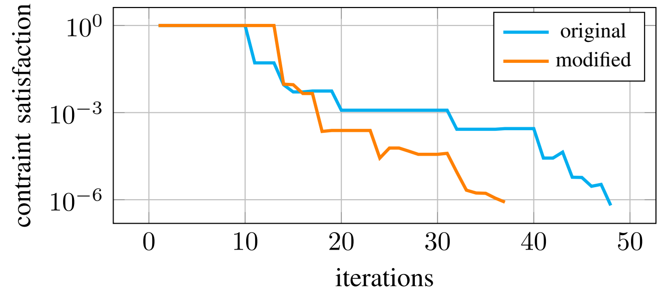

Fig. 5. Constraint convergence when solving the barrell roll problem. Compares the convergence of the original version of ALTRO versus the new, modified version that optimizing on the error state.

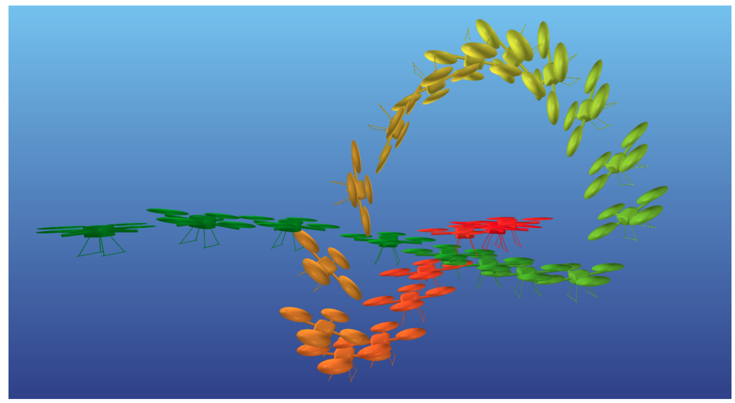

Fig. 6. Snapshots of the quadrotor flip trajectory. The red-colored quadrotors represent the state near t=0 s and the green-colored quadrotors represent the state near t=5.0 s

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 abstracted away from the user. By exploiting both the unique Markovian structure of the trajectory optimization problem and the group structure of rotations, ALTRO will likely be able to solve new classes of problems with at or near real-time performance. Future work will focus on continued refinement of the implementation and benchmarking against current state-of-the-art methods for trajectory optimization.

Additionally, 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 also include the application of these methods to multi-body robotic systems, such as humanoids or quadrupeds, especially in methods that exploit "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

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] 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.

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

[23] 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.

[24] 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.

[25] 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).

[26] 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).

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

[28] 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).

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

[30] 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.

[31] 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.

[32] 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).

[33] 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.

[34] 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.

[35] 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.