Planning with Attitude

Brian Jackson[1] and Zachary Manchester[1]

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 simply representing the attitude of these sytems is nontrivial, generating and tracking motion plans for floating-base systems is an even more challenging problem.

Many approaches have been taken to address the problem of motion planning and control on the $SE(3)$ manifold; most of these rely heavily on ideas and notation from differential geometry. Despite some impressive work in this field, these ideas have not been widely adopted and many continue to ignore the group structure of rotations, especially in the field of optimal control and motion planning. One part of this problem is that accounting for this structure requires low-level changes to the underlying optimization algorithms, which is difficult or impossible to implement when relying on commercial or even open-source optimization solvers.

In the current abstract, we formulate a straightforward, generic method for adapting existing gradient-based algorithms to correctly account for the group structure of rotations. Based entirely on basic mathematical tools from linear algebra and calculus, our method is computationally efficient and should be both accessible and familiar to most roboticists. We then apply this method to our ALTRO solver [1] and demonstrate its performance on several constrained trajectory optimization problems.

II. NOTATION AND CONVENSIONS

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 $f(x) : \mathbb{R}^n \to \mathbb{R}$, $\frac{\partial f}{\partial x} \in \mathbb{R}^{1\times n}$.

We 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. We denote the composition of two quaternions as $q_3 = q_2 \otimes q_1$.

III. QUATERNION DIFFERENTIAL CALCULUS

Most modern methods for planning and control require derivatives with respect to the state vector. We present a simple but powerful method for taking derivatives of quaternion-valued state vectors. The adaptation of this method to mixed state vectors, such as $SE(3)$, is straightforward.

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

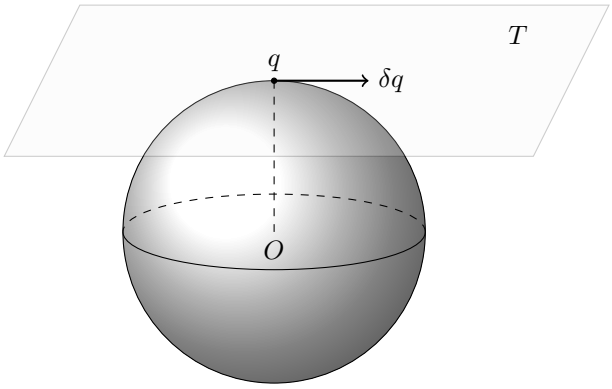

Fig. 1. When linearizing about a point q on an sphere $\mathbb{S}^{n-1}$ in n-dimensional space, the tangent space T is a plane living in $\mathbb{R}^{n-1}$, illustrated here with $n = 3$. Therefore, when linearizing about a unit quaternion $q \in \mathbb{S}^3$, the space of differential rotations lives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

The key idea of this section is that differential quaternions live in three-dimensional space instead of the four-dimensional space of quaternions. This can be seen graphically (see Figure 1) or deduced intuitively, since rotations are inherently three-dimensional and differential rotations should live in the same space as angular velocity, i.e. $\mathbb{R}^3$.

In practice we have found Rodrigues Parameters to be a very effective representation for three-dimensional differential rotations since they are computionally efficient and do not inherit the sign ambiguity of 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:

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

We will also make use of the inverse Cayley map:

$$\phi = \varphi^{-1}(q) = \frac{q_v}{q_s}. \tag{2}$$

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 unit quaternion inputs $y = h(q) : \mathbb{S}^3 \to \mathbb{R}^p$, such that

$$y + \delta y = h(q \otimes \varphi(\phi)) \approx h(q) + \nabla h(q)\phi. \tag{3}$$

We can calculate the Jacobian $\nabla h(q) \in \mathbb{R}^{p\times 3}$ by differentiating (3) with respect to ϕ, evaluated at $\phi = 0$:

$$\nabla h(q) = \frac{\partial h}{\partial q} \begin{bmatrix} -q_v^T \\ sI_3 + [q_v]_\times \end{bmatrix} := \frac{\partial h}{\partial q} G(q) \tag{4}$$

where $G(q) \in \mathbb{R}^{4\times3}$ 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 $\partial h/\partial q \in \mathbb{R}^{p\times4}$ 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. Other derivatives

Using similar techniques, we find useful expressions for the Hessian of a scalar-valued function (i.e. $p = 1$):

$$\nabla^2 h(q) = G(q)^T \frac{\partial^2 h}{\partial q^2} G(q) + I_3 \frac{\partial h}{\partial q} q, \tag{5}$$

and the Jacobian of a quaternion-valued function $q' = f(q) : \mathbb{S}^3 \to \mathbb{S}^3$:

$$\nabla f(q) = G(q')^T \frac{\partial f}{\partial q} G(q). \tag{6}$$

IV. Fast Constrained Trajectory Optimization on $SE(3)$

A. Algorithmic Modifications

Here we briefly outline the changes to the ALTRO solver [1] to efficiently solve trajectory optimization problems for systems with 3D rotations in the state vector. The linearization of the nonlinear discrete dynamics function $x_{k+1} = f(x_k, u_k)$ is "corrected" using 6:

$$A = G(f(x,u))^T \frac{\partial f}{\partial x} G(x); \quad B = G(f(x,u))^T \frac{\partial f}{\partial u}. \tag{7}$$

We can similarly compute the linearization of a nonlinear constraint function $c(x, u)$ using 4:

$$c_x = \frac{\partial c}{\partial x} G(x); \quad c_u = \frac{\partial c}{\partial x}. \tag{8}$$

Lastly, we adjust the computation of the quadratic expansion of the augmented Lagrangian cost function $J(x,u) + \lambda^T c(x,u) + \frac{1}{2} c(x,u)^T I_\mu c(x,u)$ using 4 and 5:

$$Q = G(x)^T \frac{\partial^2 J}{\partial x^2} G(x) + c_x^T I_\mu c_x + I_3 \frac{\partial J}{\partial x} x \tag{9a}$$

$$R = \frac{\partial^2 J}{\partial u^2} + c_u^T I_\mu c_u \tag{9b}$$

$$H = \frac{\partial^2 J}{\partial u \partial x} G(x) + c_u^T I_\mu c_x \tag{9c}$$

$$q = G(x)^T \frac{\partial J}{\partial x} + c_x^T (I_\mu c(x,u) + \lambda) \tag{9d}$$

$$r = \frac{\partial J}{\partial u} + c_u^T (I_\mu c(x,u) + \lambda). \tag{9e}$$

Here we define the *state attitude Jacobian* $G(x)$ to be a block-diagonal matrix where the block is an identity matrix for any vector-valued state and $G(q)$ for any quaternion in the state vector. We refer the reader to the original ALTRO paper [1] for futher details on the algorithm.

These adjustments effectively modify the optimization so that it is performed on the quaternion error state (or Rodrigues Parameters) instead of the space of unit quaternions.

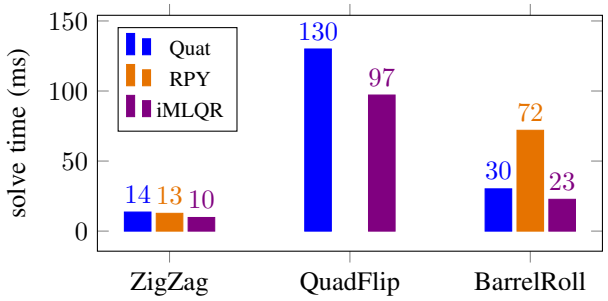

Fig. 2. Timing results for the airplane barrell roll, quadrotor flip, and quadrotor zig-zag examples using iLQR with roll-pitch-yaw Euler angles (RPY), quaternions, and the current method. The timing result for the Quadrotor flip with Euler angles is omitted since it failed to converge.

In practice, this is analagous to the Multiplicative Extended Kalman Filter [2] in the state estimation community.

The algorithm only needs one small modification to the "forward pass," where the trajectory is modified by simulating the system forward using the optimal feedback policy, $\delta u = K\delta x + d$, computed during the "backward" pass. Instead of computing δx using simple subtraction, we now compute the error state using the inverse Cayley map (2) for the quaternion states. The rest of the forward pass is unmodified.

B. Examples

To demonstrate the effectiveness of the proposed approach, we ran the modified version of ALTRO on several problems involving arbitrary 3D orientations. One required a quadrotor to do an aggresive "zig-zag" pattern, another to do a flip, and the last tasked a model airplane model to do a barrell roll. The timing results are shown in Figure 2.

V. Conclusion

We have proposed a general, accessible, and computationally efficient approach to doing planning and control for systems with one or more unit quaternions in their state vectors. The approach allows for straightforward adaptation of many gradient-based methods for optimal control and motion planning, which we demonstrated on the ALTRO solver. We have demonstrated that correctly leveraging the group structure of rotations results in lower solve times and improved robustness of the optimization algorithm.

References

[1] 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.

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