Continuous-Time Estimation of Orientation using B-Splines on SO(3)

Paul Furgale[1], Hannes Sommer[1], and James Forbes[2]
[1] ETH Zürich
[2] McGill University

1 Introduction

Discuss: How to write SO(3)? there were different versions in this paper: $SO(3)$ and SO(3) - I introduced a command and normalized to the latter...

Continuous-time estimation using temporal basis functions (proposed in Furgale et al. (2012)) enables the use of batch, maximum-likelihood estimation for problems that were previously intractable, or solvable only using filtering algorithms. In discrete time, the question of how to parameterize a three-degree-of-freedom orientation—a member of the non-commutative group SO(3) —has been subject of decades of research and debate (fill in citations). The question is no less important in continuous time, where we must decide how to parameterize a continuously time-varying orientation. Furgale et al. (2012) use the simple approach of defining a 3×1 B-spline of Cayley-Gibbs-Add "Rodrigues" to dictionary parameters (Bauchau and Trainelli, 2003). The B-spline provides analytical formulas for parameter rates, allowing the computation of angular velocity at any point Hughes (1986).

However, it is unclear if a B-spline over a minimal rotation parameterization is a good representation of orientation in continuous time. There are several issues in particular that need to be addressed. In the absence of other information, a batch estimator will produce an answer that takes the shortest distance in parameter space. This is not necessarily the same as the shortest distance in the space of rotations[1]. Because of this the estimate produced may be dependent on the coordinate frame in which we choose to express the problem, as this coordinate frame decides what part of parameter space the answer lives in. Furthermore, every minimal parameterization of rotation has a singularity and so, when using this approach, there may be a danger of approaching this singularity during the estimation process.

Kim et al. (1995) propose a method of generating B-splines on SO(3) using unit-length quaternions and their associated exponential map. The curves generated by this approach are valid unit-length quaternions at every point, and time derivatives of the curve are found by a straightforward use of the properties of the exponential map. Consequently, we would like to evaluate the approach proposed in Furgale et al. (2012) with the one proposed in Kim et al. (1995). To do so, we must further develop the theory presented in Kim et al. (1995) to be suitable for estimation. Specifically, we need the following:

1. the equation for angular velocity of a body,
2. the equation for angular acceleration of a body,
3. analytical Jacobians that relate changes in the (unit-length quaternion) spline control vertices to small changes in orientation, angular velocity, and angular acceleration, and

[1] This concept will be formalized in Section XXX

4. a method of initializing the spline control vertices to act as an initial guess for batch, nonlinear minimization.

We will develop these for both the simple approach and quaternion splines and then compare them on a simulated spacecraft-attitude-estimation problem with the following evaluation:

1. RMS error with or without a motion model constraining angular acceleration (angular acceleration = white noise)
2. computational complexity of each approach.
3. sensitivity of each approach to the choice of coordinate frame (and proof that the Kim splines are bi-invariant).

2 Related Work

3 Theory

3.1 Curves in Parameter Space

1. the equation for angular velocity of a body,
2. the equation for angular acceleration of a body,
3. analytical Jacobians that relate changes in the (unit-length quaternion) spline control vertices to small changes in orientation, angular velocity, and angular acceleration, and
4. a method of initializing the spline control vertices to act as an initial guess for batch, nonlinear minimization.

3.2 Quaternion B-Splines

1. the equation for angular velocity of a body,
2. the equation for angular acceleration of a body,
3. analytical Jacobians that relate changes in the (unit-length quaternion) spline control vertices to small changes in orientation, angular velocity, and angular acceleration, and
4. a method of initializing the spline control vertices to act as an initial guess for batch, nonlinear minimization.

Discuss: Was this meant as an table of content and should be updated or as a TODO list and thus should be removed?

In this section, we outline the approach for unit quaternion B-splines presented in Kim et al. (1995), derive the Jacobians required for state estimation, and present the equations for angular velocity and acceleration. Fundamental to the approach in Kim et al. (1995) is the use of the quaternion exponential and quaternion logarithm operations on unit length quaternions, $\left\{\mathbf{q} | \mathbf{q} \in \mathbb{R}^4, \mathbf{q}^T\mathbf{q} = 1\right\}$. We will rely on the quaternion algebra presented in Barfoot et al. (2011). In this notation, we define the components of the quaternion to be

$$\mathbf{q} =: \begin{bmatrix} \boldsymbol{\epsilon} \\ \eta \end{bmatrix}, \tag{1}$$

where $\boldsymbol{\epsilon}$ is 3×1 and η is a scalar. The quaternion *left-hand compound* operator, $(\cdot)^+$, and the *right-hand compound* operator, $(\cdot)^\oplus$, will be defined as

$$\mathbf{q}^+ := \begin{bmatrix} \eta\mathbf{1} - \boldsymbol{\epsilon}^\times & \boldsymbol{\epsilon} \\ -\boldsymbol{\epsilon}^T & \eta \end{bmatrix} \text{ and } \mathbf{q}^\oplus := \begin{bmatrix} \eta\mathbf{1} + \boldsymbol{\epsilon}^\times & \boldsymbol{\epsilon} \\ -\boldsymbol{\epsilon}^T & \eta \end{bmatrix} \tag{2}$$

where $\mathbf{1}$ is the 3×3 identity matrix.

Discuss: something strange with the sign before $\boldsymbol{\epsilon}^\times$: it would result in $\omega \sim -2\phi$

Under these definitions, the *multiplication* of quaternions, $\mathbf{q}$ and $\mathbf{r}$, which is denoted as $\mathbf{q} \cdot \mathbf{r}$ or just $\mathbf{qr}$, is equal to

$$\mathbf{q}^+\mathbf{r} \text{ or } \mathbf{r}^\oplus\mathbf{q}, \tag{3}$$

which are both products of a 4×4 matrix with a 4×1 column. The set of quaternions without zero forms with their multiplication a *non-commutative group*. Both compound operators, $(\cdot)^+$ and $(\cdot)^\oplus$, are injective homomorphisms on that multiplicative group. The former into $GL(4, \mathbb{R})$ (the group of invertible 4x4 real matrices) and the latter into $GL(4, \mathbb{R})^{\text{op}}$ (the opposite group, with reversed multiplication) (Shuster, 1993).

Discuss: I once repaired the latter statement. The former version (cited from shuster93 ?) was just broken, but most likely wanted to say something like it does now. But firstly is that still citing and secondly for what do we need that statement, anyway? For the following 'thus'?

The *identity element* of the quaternion group, $\boldsymbol{\iota} := \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$, is thus such that

$$\boldsymbol{\iota}^+ = \boldsymbol{\iota}^\oplus = \mathbf{1}, \tag{4}$$

where $\mathbf{1}$ is the 4×4 identity matrix.

None of the preceding constructs require the quaternions to be of unit length. However, given two unit-length quaternions, $\mathbf{q}$ and $\mathbf{r}$,

$$\mathbf{q}^T\mathbf{q} = 1, \quad \mathbf{r}^T\mathbf{r} = 1, \tag{5}$$

both the $(\cdot)^+$ and $(\cdot)^\oplus$ operators preserve the unit length (because the quaternion multiplication does) :

$$\left(\mathbf{q}^+\mathbf{r}\right)^T\left(\mathbf{q}^+\mathbf{r}\right) = 1, \quad \left(\mathbf{q}^\oplus\mathbf{r}\right)^T\left(\mathbf{q}^\oplus\mathbf{r}\right) = 1 \tag{6}$$

and the *conjugate* operator coincides with multiplicative inverse operator

$$\mathbf{q}^{-1} = \begin{bmatrix} -\boldsymbol{\epsilon} \\ \eta \end{bmatrix}. \tag{7}$$

The special rotation matrix, $\mathbf{C} \in \mathrm{SO}(3)$, associated with the unit length $\mathbf{q}$ by the usual equality $\forall_{\mathbf{x}\in\mathbb{R}^3}\mathbf{Cx} = \mathbf{q}^{-1}[\mathbf{x}\,0]^T\mathbf{q}$ may be built using

$$\mathbf{q}^+\mathbf{q}^{-1\oplus} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{8}$$

The Lie-Group of unit-length quaternions' Lie-Algebra consists exactly of the pure imaginary quaternions. We will represent this Lie-Algebra vectors with three vectors $\boldsymbol{\phi} \in \mathbb{R}^3$ representing the pure imaginary quaternion $[\boldsymbol{\phi}\,0]^T \in \mathbb{H}$. To convert from three to four vectors we will use the 4×3 matrix

$$\mathbf{V} := \begin{bmatrix} \mathbf{1} \\ \mathbf{0}^T \end{bmatrix}, \text{ implying } \mathbf{V}\boldsymbol{\phi} = \begin{bmatrix} \boldsymbol{\phi} \\ 0 \end{bmatrix}. \tag{9}$$

This Lie-Group's log and exp functions are this way defined on respectively into $\mathbb{R}^3$ and besides that just the restrictions of their counterparts for the full quaternions and can be calculated the following way

$$\exp(\boldsymbol{\phi}) = \begin{cases} \boldsymbol{\iota} & , \boldsymbol{\phi} = 0 \\ \begin{bmatrix} (\sin\phi)\mathbf{a} \\ \cos\phi \end{bmatrix} & , \boldsymbol{\phi} \neq 0 \end{cases}, \quad \log(\boldsymbol{q}) = \begin{cases} 0 & , \boldsymbol{q} = \boldsymbol{\iota} \\ \frac{\arccos\eta}{\sqrt{1-\eta^2}}\boldsymbol{\epsilon} & , \boldsymbol{q} \neq \pm\boldsymbol{\iota} \\ \text{undefined} & , \boldsymbol{q} = -\boldsymbol{\iota} \end{cases}, \tag{10}$$

where $\boldsymbol{\phi}$ is 3×1, $\phi := \sqrt{\boldsymbol{\phi}^T\boldsymbol{\phi}}$, and $\mathbf{a} := \boldsymbol{\phi}/\phi$. Note that

$$\log(\mathbf{q}^{-1}) = -\log(\mathbf{q}), \qquad \exp(-\boldsymbol{\phi}) = \exp(\boldsymbol{\phi})^{-1}, \tag{11}$$

3.3 Construction of Quaternion B-splines

Kim et al. (1995) define a B-spline quaternion curve based on *cumulative* B-spline basis functions. Cumulative basis functions represent an alternative but equivalent method of evaluating a B-spline function. A B-spline function of spline order m at time $t \in [t_i, t_{i+1})$, with t_i is the i-th knot, may be written for $i \geq m$ as

$$b(t) = \sum_{j=s}^{i} b_j(t)\mathbf{c}_j, \tag{12}$$

where $s := i - (m-1)$, $\mathbf{c}_j \in V$ denoting the j-th control vertex in a $\mathbb{R}$-vector space V and b_j denoting the j-th B-Spline basis function belonging to spline order m. This expression may be rearranged into the cumulative form as follows

$$b(t) = \sum_{j=s}^{i} b_j(t)\mathbf{c}_j = \mathbf{c}_s + \sum_{j=s+1}^{i} \Big(\sum_{k=j}^{i} b_k(t)\Big)(\mathbf{c}_j - \mathbf{c}_{j-1}) = \mathbf{c}_s + \sum_{j=1}^{m-1} \beta_{i,j}(t)(\mathbf{c}_j - \mathbf{c}_{j-1}), \tag{13}$$

defining the cumulative basis functions $\beta_{i,j}(t) := \sum_{k=s+j}^{i} b_k(t)$ for $1 \leq j \leq m-1$.

Discuss: the Kim definition is not segment local as our βs, which is important in his paper to make the the C^{k-2} continuity at the knots obvious. Should we mention / explain this difference in notation?

The B-spline quaternion curve in Kim et al. (1995) is built from (13) by analogy using the Lie algebra associated with the Lie group of the unit length quaternions. Rather than by vectors, $\mathbf{c}_i$, the curve is defined by a family of unit length quaternions control vertices, $\mathbf{q}_i$. The quaternion equivalent of (13) becomes

$$\mathbf{q}(t) := \mathbf{q}_s \prod_{j=1}^{m-1} \mathbf{r}_{i,j}(t), \tag{14}$$

where

$$\mathbf{r}_{i,j}(t) := \exp\left(\beta_{i,j}(t)\boldsymbol{\varphi}_{s+j}\right), \tag{15}$$

and

$$\boldsymbol{\varphi}_k := \log\left(\mathbf{q}_{k-1}^{-1}\mathbf{q}_k\right). \tag{16}$$

Defining a curve to represent SO(3) in this way has two properties that make it desirable: (a) it has no singularities, and (b) it satisfies the property of *bi-invariance* as described in Park and Ravani (1997).

Hannes: Insert proof - here or in the appendix

To use these curves for estimation, we require several pieces not derived in Kim et al. (1995), Jacobians and the equations for angular velocity and angular acceleration.

3.4 Derivatives of Quaternion B-splines

3.4.1 Time Derivatives

We can calculate the B-Splines derivatives with respect to t using the product rule, which holds for quaternion functions as usual.

$$\frac{d^k}{dt^k}\mathbf{q}(t) = \frac{d^k}{dt^k}\mathbf{q}_s \prod_{j=1}^{m-1} \mathbf{r}_{i,j}(t) = k!\mathbf{q}_s \sum_{\substack{\boldsymbol{\alpha}\in\mathbb{N}^{m-1} \\ \sum\boldsymbol{\alpha}=k}} \prod_{j=1}^{m-1} \frac{1}{\boldsymbol{\alpha}_j!} \frac{d^{\boldsymbol{\alpha}_j}}{dt^{\boldsymbol{\alpha}_j}} \mathbf{r}_{i,j}(t) \tag{17}$$

This requires the derivatives of $\mathbf{r}_{i,j}$ up to k. We only calculate those up to $k = 2$, because we luckily won't need more. In order to save the calculation of the whole differential of the exponential map here, we introduce the following directional exponential for any $\mathbf{q} \in \mathbb{H}$ as function of $t \in \mathbb{R}$:

$$\exp^{\mathbf{q}}(t) := \exp(t\mathbf{q}) \tag{18}$$

For its derivative we will benefit from the well known property of the exponential map on the quaternions (it holds just as for the matrix exponential map):

$$\frac{d}{dt}\exp^{\mathbf{q}}(t) = \mathbf{q}\exp^{\mathbf{q}}(t) \tag{19}$$

Using the directional exponential map we can easily derive expressions for the derivatives of the $\mathbf{r}_{i,j}(t)$ expressions.

$$\frac{d}{dt}\mathbf{r}_{i,j}(t) = \frac{d}{dt}\exp^{\boldsymbol{\varphi}_{s+j}}(\beta_{i,j}(t)) = \beta'_{i,j}(t)\boldsymbol{\varphi}_{s+j}\exp^{\boldsymbol{\varphi}_{s+j}}(\beta_{i,j}(t)) = \beta'_{i,j}(t)\boldsymbol{\varphi}_{s+j}\mathbf{r}_{i,j}(t) \tag{20}$$

$$\frac{d^2}{dt^2}\mathbf{r}_{i,j}(t) = (\beta'_{i,j}(t)^2\boldsymbol{\varphi}_{s+j} + \beta''_{i,j}(t)\boldsymbol{\iota})\boldsymbol{\varphi}_{s+j}\mathbf{r}_{i,j}(t) \tag{21}$$

3.4.2 Angular velocity

Assuming a unit quaternion curve $\mathbf{q}(t) = [\boldsymbol{\epsilon}(t)\ \eta(t)]^T$ - interpreted as curve through SO(3) the resulting angular velocity of a rotated frame holds (s. [?]):

$$\boldsymbol{\omega} = 2(\eta\dot{\boldsymbol{\epsilon}} - \dot{\eta}\boldsymbol{\epsilon} - \boldsymbol{\epsilon}\times\dot{\boldsymbol{\epsilon}}) \tag{22}$$

3.4.3 Angular acceleration

Under the same assumptions as for the angular velocity it follows as expression for the angular acceleration :

$$\boldsymbol{\alpha} = \dot{\boldsymbol{\omega}} = 2(\eta\ddot{\boldsymbol{\epsilon}} - \ddot{\eta}\boldsymbol{\epsilon} - \boldsymbol{\epsilon}\times\ddot{\boldsymbol{\epsilon}}) \tag{23}$$

Discuss: derivation needed? It's short and rather simple ...

3.4.4 Jacobians of Quaternion B-splines

The Levenberg Marquardt method is the predominant method of estimation in robotics. To implement this method analytically, we must have access to an expression for the Jacobian (with respect to small changes in the state variables) for any nonlinear function that appears in our error terms. In this case, the state variables are the unit-length quaternion-valued control vertex. Unit-length quaternions use four parameters for three degrees of freedom, but rather than performing constrained estimation, it is usual to use a *minimal perturbation* in the tangent space mapped to the manifold via its exponential map, when using quaternions in unconstrained estimation, according to:

$$\mathbf{q}(\delta\boldsymbol{\phi}) = \exp(\delta\boldsymbol{\phi})\overline{\mathbf{q}}, \tag{24}$$

where $\overline{\mathbf{q}}$ is our current guess, and $\delta\boldsymbol{\phi}$ is a minimal, 3×1, perturbation. When an iteration of Levenberg-Marquardt produces an answer for $\delta\boldsymbol{\phi}$, the update is applied as

$$\overline{\mathbf{q}} \leftarrow \exp(\delta\boldsymbol{\phi})\overline{\mathbf{q}}. \tag{25}$$

This update is *constraint sensitive* in that the updated $\overline{\mathbf{q}}$ is still of unit length. We would like to use the same method to perform unconstrained optimization using quaternion curves. The question is, what is the Jacobian of (14) with respect to small changes in the individual control vertices?

So let's derive the partial derivative of the whole spline with respect to the small perturbations $\boldsymbol{\phi}_l$ at zero while considering the control vertices $\mathbf{q}_l(\boldsymbol{\phi}_l) = \exp(\boldsymbol{\phi}_l)\bar{\mathbf{q}}_l$ as functions of $\boldsymbol{\phi}_l$ and omitting the time parameter:

$$\frac{\partial \mathbf{q}}{\partial \boldsymbol{\phi}_l} = \frac{\partial}{\partial \boldsymbol{\phi}_l}\mathbf{q}_s \prod_{j=1}^{m-1} \mathbf{r}_{i,j} = \left(\prod_{j=1}^{m-1} \mathbf{r}_{i,j}\right)^{\oplus} \frac{\partial \mathbf{q}_s}{\partial \boldsymbol{\phi}_l} + \mathbf{q}_s^{+} \sum_{k=1}^{m-1} \left(\prod_{j=1}^{k-1} \mathbf{r}_{i,j}\right)^{+} \left(\prod_{j=k+1}^{m-1} \mathbf{r}_{i,j}\right)^{\oplus} \frac{\partial \mathbf{r}_{i,k}}{\partial \boldsymbol{\phi}_l} \tag{26}$$

For the first summand we have $\frac{\partial \mathbf{q}_s}{\partial \boldsymbol{\phi}_l} = \delta_{sl}\mathbf{q}_s^{\oplus}\mathbf{V}$. For the derivatives of the $\mathbf{r}_{i,k}$ recalling their definition we firstly note that they vanish for any $l \notin \{k+s, k-1+s\}$. So the whole upper expression consists only of up to three non zero summands (depending on the relation of l and s). To further derive the $\mathbf{r}_{i,j}$ derivatives, we require some intermediate results. First, we define the Jacobian of the $\log(\cdot)$ function with respect to perturbations of the form in (24),

$$\log\left(\exp(\delta\boldsymbol{\phi})\overline{\mathbf{q}}\right) \approx \log(\overline{\mathbf{q}}) + \mathbf{L}(\overline{\mathbf{q}})\delta\boldsymbol{\phi}, \tag{27}$$

where

$$\mathbf{L}(\mathbf{q}) := \begin{cases} \mathbf{1} & ,\mathbf{q} = \boldsymbol{\iota} \\ \mathbf{1} + \boldsymbol{\phi}^{\times} + \left(1 - \frac{\phi}{\tan\phi}\right)\mathbf{a}^{\times}\mathbf{a}^{\times} & ,\mathbf{q} \neq \boldsymbol{\iota} \end{cases}, \tag{28}$$

and we have used $\boldsymbol{\phi} := \log(\mathbf{q})$, $\phi := \sqrt{\boldsymbol{\phi}^T\boldsymbol{\phi}}$, and $\mathbf{a} := \boldsymbol{\phi}/\phi$. Next, we need the Jacobian of the $\exp(\cdot)$ function with respect to perturbations in $\boldsymbol{\theta}$. This can be written as

$$\exp\left(\overline{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}\right) \approx \left(\boldsymbol{\iota} + \mathbf{VS}\left(\overline{\boldsymbol{\theta}}\right)\delta\boldsymbol{\theta}\right)^{+} \exp\left(\overline{\boldsymbol{\theta}}\right) = \exp\left(\overline{\boldsymbol{\theta}}\right) + \exp\left(\overline{\boldsymbol{\theta}}\right)^{\oplus} \mathbf{VS}\left(\overline{\boldsymbol{\theta}}\right)\delta\boldsymbol{\theta}, \tag{29}$$

where

$$\mathbf{S}(\boldsymbol{\theta}) := \mathbf{1} - \frac{1}{\theta}\sin^2\theta\,\mathbf{a}^{\times} + \frac{1}{\theta}(\theta - \frac{1}{2}\sin 2\theta)\mathbf{a}^{\times}\mathbf{a}^{\times}. \tag{30}$$

We note, without proof, that $\mathbf{L}(\mathbf{q}) = \mathbf{S}(\log(\mathbf{q}))^{-1}$.

Discuss: For what the last sentence? I think, we don't need it.

Next, we see how perturbations of the form (24) become perturbations in φ_k from (16):

$$\begin{aligned}
\varphi_k &= \log\left(\mathbf{q}_{k-1}^{-1}\mathbf{q}_k\right) && (31a)\\
&= \log\left(\overline{\mathbf{q}}_{k-1}^{-1}\left(\boldsymbol{\iota} - \mathbf{V}\delta\boldsymbol{\phi}_{k-1}\right)\left(\boldsymbol{\iota} + \mathbf{V}\delta\boldsymbol{\phi}_k\right)\overline{\mathbf{q}}_k\right) && (31b)\\
&\approx \log\left(\overline{\mathbf{q}}_{k-1}^{-1}\overline{\mathbf{q}}_k - \overline{\mathbf{q}}_{k-1}^{-1}(\mathbf{V}\delta\boldsymbol{\phi}_{k-1})\overline{\mathbf{q}}_k + \overline{\mathbf{q}}_{k-1}^{-1}(\mathbf{V}\delta\boldsymbol{\phi}_k)\overline{\mathbf{q}}_k\right) && (31c)\\
&= \log\left(\overline{\mathbf{q}}_{k-1}^{-1}\overline{\mathbf{q}}_k - (\mathbf{V}\overline{\mathbf{C}}_{k-1}^T\delta\boldsymbol{\phi}_{k-1})\overline{\mathbf{q}}_{k-1}^{-1}\overline{\mathbf{q}}_k + (\mathbf{V}\overline{\mathbf{C}}_{k-1}^T\delta\boldsymbol{\phi}_k)\overline{\mathbf{q}}_{k-1}^{-1}\overline{\mathbf{q}}_k\right) && (31d)\\
&= \log\left(\left(\iota + \left(\mathbf{V}\overline{\mathbf{C}}_{k-1}^T\delta\boldsymbol{\phi}_k - \mathbf{V}\mathbf{C}_{k-1}^T\delta\boldsymbol{\phi}_{k-1}\right)\right)\overline{\mathbf{q}}_{k-1}^{-1}\overline{\mathbf{q}}_k\right) && (31e)\\
&= \log\left(\left(\iota + \left(\mathbf{V}\begin{bmatrix}\overline{\mathbf{C}}_{k-1}^T & -\overline{\mathbf{C}}_{k-1}^T\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{\phi}_k\\ \delta\boldsymbol{\phi}_{k-1}\end{bmatrix}\right)\right)\overline{\mathbf{q}}_{k-1}^{-1}\overline{\mathbf{q}}_k\right) && (31f)\\
&\approx \log\left(\overline{\mathbf{q}}_{k-1}^{-1}\overline{\mathbf{q}}_k\right) + \mathbf{L}\left(\overline{\mathbf{q}}_{k-1}^{-1}\overline{\mathbf{q}}_k\right)\begin{bmatrix}\overline{\mathbf{C}}_{k-1}^T & -\overline{\mathbf{C}}_{k-1}^T\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{\phi}_k\\ \delta\boldsymbol{\phi}_{k-1}\end{bmatrix} && (31g)
\end{aligned}$$

In the above derivation we used many identities from the quaternion algebra presented in Barfoot et al. (2011). We have also defined $\overline{\mathbf{C}}_{k-1}$ to be the rotation matrix built from the quaternion $\overline{\mathbf{q}}_{k-1}$. The trickiest manipulation was between (31c) and (31d). The sub expressions were simplified as follows (subscripts omitted for clarity):

$$\begin{aligned}
\mathbf{q}(\mathbf{V}\delta\boldsymbol{\phi}) &= \mathbf{q}(\mathbf{V}\delta\boldsymbol{\phi})\underbrace{\mathbf{q}^{-1}\mathbf{q}}_{\boldsymbol{\iota}} && (32a)\\
&= \left(\mathbf{q}^{+}\mathbf{q}^{-1\oplus}\mathbf{V}\delta\boldsymbol{\phi}\right)\mathbf{q} && (32b)\\
&= \left(\begin{bmatrix}\mathbf{C} & \mathbf{0}\\ \mathbf{0}^T & 1\end{bmatrix}\begin{bmatrix}\mathbf{1}\\ \mathbf{0}^T\end{bmatrix}\delta\boldsymbol{\phi}\right)\mathbf{q} && (32c)\\
&= \left(\begin{bmatrix}\mathbf{C}\\ \mathbf{0}^T\end{bmatrix}\delta\boldsymbol{\phi}\right)\mathbf{q} && (32d)\\
&= (\mathbf{V}\mathbf{C}\delta\boldsymbol{\phi})\,\mathbf{q} && (32e)
\end{aligned}$$

In the above, $\mathbf{C}$ is the rotation matrix built from the quaternion, $\mathbf{q}$. Altogether we finally get for any $1 \leq j < m$ and $k := s + j$

$$\frac{\partial \mathbf{r}_{i,j}}{\partial \boldsymbol{\phi}_k} = \frac{\partial}{\partial \boldsymbol{\phi}_k}\exp\left(\beta_{i,j}\log\left(\mathbf{q}_{k-1}^{-1}\mathbf{q}_k\right)\right) = \exp(\beta_{i,j}\overline{\boldsymbol{\varphi}}_k)^{\oplus}\mathbf{V}\mathbf{S}(\beta_{i,j}\overline{\boldsymbol{\varphi}}_k)\beta_{i,j}\mathbf{L}\left(\overline{\mathbf{q}}_{k-1}^{-1}\overline{\mathbf{q}}_k\right)\overline{\mathbf{C}}_{k-1}^T \tag{33}$$

and

$$\frac{\partial \mathbf{r}_{i,j}}{\partial \boldsymbol{\phi}_{k-1}} = -\frac{\partial \mathbf{r}_k}{\partial \boldsymbol{\phi}_k} \tag{34}$$

We will also need the corresponding Jacobians for the time derivatives of the B-Spline and their derived quantities. Considering (26) and (34) we have to essentially derive the following - not expanding the derivatives of $\boldsymbol{\varphi}_k$ for the sake of readability:

$$\frac{\partial}{\partial \boldsymbol{\phi}_k}\frac{d}{dt}\mathbf{r}_{i,j}(t) = \beta'_{i,j}(t)\frac{\partial}{\partial \boldsymbol{\phi}_k}(\mathbf{V}\boldsymbol{\varphi}_k)\mathbf{r}_{i,j}(t) = \beta'_{i,j}(t)\left(\mathbf{r}_{i,j}(t)^{\oplus}\mathbf{V}\frac{\partial \boldsymbol{\varphi}_k}{\partial \boldsymbol{\phi}_k} + (\mathbf{V}\boldsymbol{\varphi}_k)\frac{\partial \mathbf{r}_{i,j}(t)}{\partial \boldsymbol{\phi}_k}\right) \tag{35}$$

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\phi}_j}\frac{d^2}{dt^2}\mathbf{r}_{i,j}(t) &= \frac{\partial}{\partial \boldsymbol{\phi}_k}(\beta'_{i,j}(t)^2\mathbf{V}\boldsymbol{\varphi}_k + \beta''_{i,j}(t)\boldsymbol{\iota})(\mathbf{V}\boldsymbol{\varphi}_k)\mathbf{r}_{i,j}(t)\\
&= \left(\beta'_{i,j}(t)^2\mathbf{V}\boldsymbol{\varphi}_k + \beta''_{i,j}(t)\boldsymbol{\iota}\right)^{+}\left(\mathbf{r}_{i,j}(t)^{\oplus}\mathbf{V}\frac{\partial \boldsymbol{\varphi}_k}{\partial \boldsymbol{\phi}_k} + (\mathbf{V}\boldsymbol{\varphi}_k)^{+}\frac{\partial \mathbf{r}_{i,j}(t)}{\partial \boldsymbol{\phi}_k}\right)\\
&\quad + ((\mathbf{V}\boldsymbol{\varphi}_k)\mathbf{r}_{i,j}(t))^{\oplus}(\beta'_{i,j}(t)^2\mathbf{V}\frac{\partial \boldsymbol{\varphi}_k}{\partial \boldsymbol{\phi}_k}) \qquad (36)
\end{aligned}$$

3.4.5 Angular velocity

The Jacobians of the angular velocity with respect to minimal perturbations of the control vertices are also required. So let again $1 \leq j < m$ and $k := s + j$

$$\frac{1}{2}\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{\phi}_k} = \frac{\partial \eta}{\partial \boldsymbol{\phi}_k}\dot{\boldsymbol{\epsilon}} + \eta \frac{\partial \dot{\boldsymbol{\epsilon}}}{\partial \boldsymbol{\phi}_k} - \frac{\partial \dot{\eta}}{\partial \boldsymbol{\phi}_k}\boldsymbol{\epsilon} - \dot{\eta}\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{\phi}_k} - \frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{\phi}_k} \times \dot{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon} \times \frac{\partial \dot{\boldsymbol{\epsilon}}}{\partial \boldsymbol{\phi}_k} \tag{37}$$

Hannes: Add Jacobians derivation?

where the two $\times$-operators are formally generalized to one three vector and one three by something matrix argument by applying the cross product column-wise: Let $\mathbf{A} = [\mathbf{A}_j]_{j=1}^{\mu} \in \mathbb{R}^{3\times\mu}$, where $\mathbf{A}_j$ denotes the j-th column of $\mathbf{A}$ and $\mu \in \mathbb{N}$ and define

$$\mathbf{y} \times \mathbf{A} := [\mathbf{y} \times \mathbf{A}_j]_{j=1}^{\mu}, \mathbf{A} \times \mathbf{y} := [\mathbf{A}_j \times \mathbf{y}]_{j=1}^{\mu}$$

3.4.6 Angular acceleration

For the angular acceleration it yields analogously:

$$\frac{1}{2}\frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{\phi}_k} = \frac{\partial \eta}{\partial \boldsymbol{\phi}_k}\ddot{\boldsymbol{\epsilon}} + \eta \frac{\partial \ddot{\boldsymbol{\epsilon}}}{\partial \boldsymbol{\phi}_k} - \frac{\partial \ddot{\eta}}{\partial \boldsymbol{\phi}_k}\boldsymbol{\epsilon} - \ddot{\eta}\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{\phi}_k} - \frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{\phi}_k} \times \ddot{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon} \times \frac{\partial \ddot{\boldsymbol{\epsilon}}}{\partial \boldsymbol{\phi}_k} \tag{38}$$

4 Experiments

In this section we apply the state representation derived above to two spacecraft attitude estimation problems. The goal is to estimate the attitude of a coordinate frame attached to the vehicle body, $\underrightarrow{\mathcal{F}}_b$, with respect to an inertial frame, $\underrightarrow{\mathcal{F}}_i$, over a time interval of interest, $T = [t_0, t_K]$. In the notation of the previous sections, we would like to estimate $\mathbf{q}_{ib}(t)$ over T.

In both problems, the vehicle has a bearing sensor that takes measurements at discrete time instances, $\{t_m | m = 1 \ldots M\}$. The bearing sensor is rigidly attached to the vehicle body at coordinate frame $\underrightarrow{\mathcal{F}}_s$. We assume that the orientation of the sensor frame with respect to the body frame, $\mathbf{C}_{sb}$, is known. We assume that the beacon being observed by the sensor is very far away from the vehicle (such as the Sun or another star). Let $\mathbf{b}_i^{\ell}$ be the bearing of beacon ℓ, expressed in the inertial frame. A measurement of beacon ℓ at time m, written $\mathbf{y}_{m\ell}$, is modelled as

$$\mathbf{y}_{m\ell} = \mathbf{h}\left(\mathbf{C}_{sb}\boldsymbol{\mathcal{C}}^T\left(\mathbf{q}_{ib}(t_m)\right)\mathbf{b}_i^{\ell}\right) + \mathbf{n}_{m\ell}, \qquad \mathbf{n}_{m\ell} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{m\ell}\right), \tag{39}$$

where the bearing vector is first rotated into the sensor frame, $\mathbf{h}(\cdot)$ is a nonlinear observation model, and $\mathbf{n}_{m\ell}$ is zero-mean Gaussian noise with covariance $\mathbf{R}_{m\ell}$. Following the standard practice of maximum likelihood estimation, we define the error term associated with this measurement to be

$$\mathbf{e}_{m\ell} := \mathbf{y}_{m\ell} - \mathbf{h}\left(\mathbf{C}_{sb}\boldsymbol{\mathcal{C}}^T\left(\mathbf{q}_{ib}(t_m)\right)\mathbf{b}_i^{\ell}\right), \tag{40}$$

where $\boldsymbol{\mathcal{C}}$ is the function that builds a rotation matrix from a quaternion. These errors contribute to the term J_y in our overall objective function,

$$J_y := \frac{1}{2}\sum_{m=1}^{M} \mathbf{e}_{m\ell}^T \mathbf{R}_{m\ell}^{-1} \mathbf{e}_{m\ell}. \tag{41}$$

Although both experiments use this common setup, they differ in terms of the dynamics model used. In Experiment 1, we use a dynamics model based on Euler's equation and the vehicle inertia matrix. In Experiment 2, we use measured angular velocities from a gyroscope.

4.1 Experiment 1

In Experiment 1, we consider a spacecraft attitude estimation problem that uses a vehicle dynamics model based on Euler's equation. The dynamics model is

$$\mathbf{I}\dot{\boldsymbol{\omega}}(t) + \boldsymbol{\omega}^{\times}(t)\mathbf{I}\boldsymbol{\omega}(t) = \mathbf{u}(t) + \mathbf{w}(t), \tag{42}$$

where t is time, $\mathbf{I}$ is the vehicle inertia matrix, $\boldsymbol{\omega}(t)$ is the angular velocity of the vehicle's center of mass as seen from the inertial frame, $\mathbf{u}(t)$ is a known control input, and $\mathbf{w}(t)$ is a zero-mean white Gaussian process with covariance $\mathbf{Q}_d$,

$$\mathbf{w}(t) \sim \mathcal{GP}\left(\mathbf{0}, \delta(t-t')\mathbf{Q}_d\right), \tag{43}$$

where $\delta(\cdot)$ is Dirac's delta function. Following Furgale et al. (2012), we define the error for the dynamics model at time t to be

$$\mathbf{e}_d(t) := \mathbf{I}\dot{\boldsymbol{\omega}}(t) + \boldsymbol{\omega}^{\times}(t)\mathbf{I}\boldsymbol{\omega}(t) - \mathbf{u}(t). \tag{44}$$

The resulting term in our objective function is

$$\mathbf{J}_d := \frac{1}{2}\int_{t_0}^{t_K} \mathbf{e}_d^T(t)\mathbf{Q}_d^{-1}\mathbf{e}_d(t)\,dt. \tag{45}$$

Together, (41) and (45) define the combined objective function that we would like to minimize, $J := J_y + J_d$. Let $\boldsymbol{\mathcal{Q}}$ be the stacked matrix of quaternion control vertices. Our goal is to estimate

$$\boldsymbol{\mathcal{Q}}^\star = \underset{\boldsymbol{\mathcal{Q}}}{\operatorname{argmin}}\, J. \tag{46}$$

4.2 Experiment 2

In Experiment 2, we consider an estimation problem that neglects analytical vehicle dynamics in favor of measured dynamics from a three-axis gyroscope. For simplicity, we choose to place our vehicle body frame at the center of our gyroscope measurement frame. The gyroscope measurement model follows the one commonly used in robotics (Mirzaei and Roumeliotis, 2008; Kelly and Sukhatme, 2011),

$$\boldsymbol{\varpi}_g = \boldsymbol{\mathcal{C}}^T\left(\mathbf{q}_{ib}\right)\boldsymbol{\omega}(t_g) + \mathbf{b}(t_g) + \mathbf{n}_{g\omega}, \qquad \mathbf{n}_{g\omega} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{g\omega}), \tag{47a}$$

$$\dot{\mathbf{b}}(t) = \mathbf{w}_b(t), \qquad \mathbf{w}_b(t) \sim \mathcal{GP}\left(\mathbf{0}, \delta(t-t'\mathbf{Q}_b)\right). \tag{47b}$$

In these equations, the gyroscope the G gyroscope measurements are assumed to be subject to both zero-mean Gaussian measurement noise, $\mathbf{n}_{g\omega}$, and a slowly evolving bias, $\mathbf{b}(t)$[2]. We model the bias as a B-spline function, $\mathbf{b}(t) := \boldsymbol{\Phi}_b(t)\mathbf{c}_b$, where $\boldsymbol{\Phi}_b(\cdot)$ is the stacked matrix of known B-spline basis functions and $\mathbf{c}_b$ is the unknown coefficient vector. The error for a single gyroscope measurement may be written as

$$\mathbf{e}_{g\omega} := \boldsymbol{\varpi}_g - \boldsymbol{\mathcal{C}}^T\left(\mathbf{q}_{ib}\right)\boldsymbol{\omega}(t_g) - \mathbf{b}(t_g), \tag{48}$$

which becomes a term, J_ω, in our objective function:

$$J_\omega := \frac{1}{2}\sum_{g=1}^{G}\mathbf{e}_{g\omega}^T\mathbf{R}_{g\omega}^{-1}\mathbf{e}_{g\omega}\,. \tag{49}$$

The bias motion error may be written as

$$\mathbf{e}_b(t) := \dot{\mathbf{b}}(t). \tag{50}$$

[2]The noise characteristics of a particular gyroscope may be identified using the method described in IEEE Aerospace and Electronic Systems Society. Gyro and Accelerometer Panel and Institute of Electrical and Electronics Engineers (1998)

This error contributes to our objective function as

$$J_b := \frac{1}{2} \int_{t_0}^{t_K} \mathbf{e}_b^T(t) \mathbf{Q}_b^{-1} \mathbf{e}_b(t) \, dt. \tag{51}$$

Together, (41), (49), and (51) define the combined objective function that we would like to minimize, $J := J_y + J_\omega + J_b$. Let $\boldsymbol{\mathcal{Q}}$ be the stacked matrix of quaternion control vertices. Our goal is to estimate

$$\{\boldsymbol{\mathcal{Q}}^\star, \mathbf{c}_b^\star\} = \operatorname*{argmin}_{\boldsymbol{\mathcal{Q}}, \mathbf{c}_b} J\,. \tag{52}$$

References

Barfoot, T. D., Forbes, J. R., and Furgale, P. T. (2011). Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica*, 68(1-2):101–112.

Bauchau, O. and Trainelli, L. (2003). The vectorial parameterization of rotation. *Nonlinear dynamics*, 32(1):71–92.

Furgale, P. T., Barfoot, T. D., and Sibley, G. (2012). Continuous-time batch estimation using temporal basis functions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2088–2095, St. Paul, MN.

Hughes, P. C. (1986). *Spacecraft Attitude Dynamics*. John Wiley & Sons, New York.

IEEE Aerospace and Electronic Systems Society. Gyro and Accelerometer Panel and Institute of Electrical and Electronics Engineers (1998). *IEEE Standard Specification Format Guide and Test Procedure for Single-axis Interferometric Fiber Optic Gyros*. IEEE (std.). IEEE.

Kelly, J. and Sukhatme, G. S. (2011). Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1):56–79.

Kim, M.-J., Kim, M.-S., and Shin, S. Y. (1995). A general construction scheme for unit quaternion curves with simple high order derivatives. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 369–376, New York, NY, USA. ACM.

Mirzaei, F. and Roumeliotis, S. (2008). A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *Robotics, IEEE Transactions on*, 24(5):1143–1156.

Park, F. C. and Ravani, B. (1997). Smooth invariant interpolation of rotations. *ACM Trans. Graph.*, 16:277–295.

Shuster, M. D. (1993). A survey of attitude representations. *Journal of the Astronautical Sciences*, 41(4):439–517.