Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2013-005 March 27, 2013

Tracking 3-D Rotations with the Quaternion Bingham Filter

Jared Glover and Leslie Pack Kaelbling

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Tracking 3-D Rotations with the Quaternion Bingham Filter

Jared Glover and Leslie Pack Kaelbling
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
`jglov|lpk@mit.edu`

March 27, 2013

Abstract

A deterministic method for sequential estimation of 3-D rotations is presented. The Bingham distribution is used to represent uncertainty directly on the unit quaternion hypersphere. Quaternions avoid the degeneracies of other 3-D orientation representations, while the Bingham distribution allows tracking of large-error (high-entropy) rotational distributions. Experimental comparison to a leading EKF-based filtering approach on both synthetic signals and a ball-tracking dataset shows that the Quaternion Bingham Filter (QBF) has lower tracking error than the EKF, particularly when the state is highly dynamic. We present two versions of the QBF–suitable for tracking the state of first- and second-order rotating dynamical systems.

1 Introduction

3-D rotational data occurs in many disciplines, from geology to robotics to physics. Yet modern statistical inference techniques are seldom applied to such data sets, due to the complex topology of 3-D rotation space, and the well-known aliasing problems caused by orientations "wrapping around" back to zero. Many probability distributions exist for modeling uncertainty on rotational data, yet difficulties often arise in the mechanics of complex inference tasks. The goal of this paper is to explore one distribution—the Bingham—which is particularly well-suited for inference, and to derive many common operations on Bingham distributions as a reference for future researchers.

We present a new deterministic method—the Quaternion Bingham Filter (QBF)—for approximate recursive inference in quaternion Bingham processes. The quaternion Bingham process is a type of dynamic Bayesian network (DBN) on 3-D rotational data, where both process dynamics and measurements are perturbed by random Bingham rotations. The QBF uses the Bingham distribution to represent state uncertainty on the unit quaternion hypersphere. Quaternions avoid the degeneracies of other 3-D orientation representations, while the Bingham distribution enables accurate tracking of high-noise signals. Performing exact inference on quaternion Bingham processes requires the composition of Bingham distributions, which results in a non-Bingham density. Therefore, we approximate the resulting composed distribution as a Bingham using the method of moments, in order to keep the filtered state distribution in the Bingham family.

We compare the quaternion Bingham filter to a previous approach to tracking rotations based on the Extended Kalman Filter (EKF) and find that the QBF has lower tracking error than the EKF, particularly when the state is highly dynamic. We evaluate the performance of the QBF on both synthetic rotational process signals and on a real dataset containing 3-D orientation estimates of a spinning ping-pong ball tracked in high-speed video. We also derive the true probability density function (PDF) for the composition of two Bingham distributions, and report the empirical error of the moment-matching composition approximation for various distributions. We begin by introducing the Bingham distribution along with the first- and second-order quaternion Bingham processes and the QBF for estimating their state. We then describe some important operations on the Bingham distribution, and conclude with experiments on artificial and real data.

2 Distributions on rotations

The problem of how to represent a probability distribution on the space of rotations in three dimensions has been a subject of considerable study. Representing the distribution directly in the space of Euler angles is difficult because of singularities in the space when two of the angles are aligned (known as gimbal lock). A more appropriate space for representing distributions on rotations is the space of unit quaternions: a rotation becomes a point on the 4-dimensional unit hypersphere, $\mathbb{S}^3$. This space lacks singularities, but has the difficulty that the representation is not unique: both $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. Putting a Gaussian distribution directly in quaternion space does not respect the underlying topology of 3-D rotations; however, this approach has been the basis of tracking methods based on approximations of the Kalman filter [13, 3, 15, 4, 9, 10]. A more appropriate method is to represent distributions in an $\mathbb{R}^3$ space that is tangent to the quaternion hypersphere at the mean of the distribution [6]; but such a tangent-space approach will be unable to effectively represent distributions that have large variances. In many perceptual problems, it may be possible to make observations that provide significant information about only one or two dimensions, yielding high-variance estimates. For this reason, we use the Bingham distribution.

The Bingham distribution [1] is an antipodally symmetric probability distribution on a unit hypersphere. It can be used to represent many types of uncertainty, from highly peaked distributions to distributions which are symmetric about some axis to the uniform distribution. It is thus a very flexible distribution for unit quaternions on the hypersphere $\mathbb{S}^3$. Despite its versatility, it has not yet seen wide usage in the AI and robotics communities. This is because the normalization constant in the Bingham density cannot be computed in closed form, which makes many types of (exact) inference difficult. This difficulty can be overcome using caching techniques and interpolation to approximate the normalization constant, F, and its partial derivatives, ∇F, with respect to the Bingham concentration parameters, Λ.

The probability density function (PDF) for the Bingham distribution is

$$f(\mathbf{x};\Lambda,V) = \frac{1}{F}\exp\{\sum_{i=1}^{d}\lambda_i(\mathbf{v_i}^T\mathbf{x})^2\} \tag{1}$$

where $\mathbf{x}$ is a unit vector on the surface of the sphere $\mathbb{S}^d \subset \mathbb{R}^{d+1}$, F is a normalization constant, Λ is a vector of concentration parameters, and the columns of the $(d+1)\times d$ matrix V are orthogonal unit vectors. We refer to the density in equation 6 as the *standard form* for the Bingham. A more succinct form for the Bingham is $f(\mathbf{x};C) = \frac{1}{F}\exp(\mathbf{x}^T C\mathbf{x})$, where C is a $(d+1)\times(d+1)$ orthogonal matrix; however, C is clearly an over-parameterization since the first d columns of C completely determine its final column. By convention, one typically defines Λ and V so that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_d \leq 0$. Note that a large-magnitude λ_i indicates that the distribution is highly peaked along the direction $\mathbf{v_i}$, while a small-magnitude λ_i indicates that the distribution is spread out along $\mathbf{v_i}$.

The Bingham distribution is derived from a zero-mean Gaussian on $\mathbb{R}^{d+1}$, conditioned to lie on the surface of the unit hypersphere $\mathbb{S}^d$. Thus, the exponent of the Bingham PDF is the same as the exponent of a zero-mean Gaussian distribution (in principal components form, with one of the eigenvalues of the covariance matrix set to infinity). The Bingham distribution is the *maximum entropy* distribution on the hypersphere which matches the sample inertia matrix $E[\mathbf{x}\mathbf{x}^T]$ [14]. Therefore, it may be better suited to representing random process noise on the hypersphere than some other distributions, such as (projected) tangent-space Gaussians.

The Bingham distribution is most commonly used to represent uncertainty in axial data on the sphere, $\mathbb{S}^2$. In geology, it is often used to encode preferred orientations of minerals in rocks [12, 16]. Higher dimensional, complex forms of the Bingham distribution are also used to represent the distribution over 2-D planar shapes [5]. In this work, we use the Bingham on $\mathbb{S}^3$ as a probability distribution over 3-D quaternion rotations. Since the unit quaternions $\mathbf{q}$ and $-\mathbf{q}$ represent the same

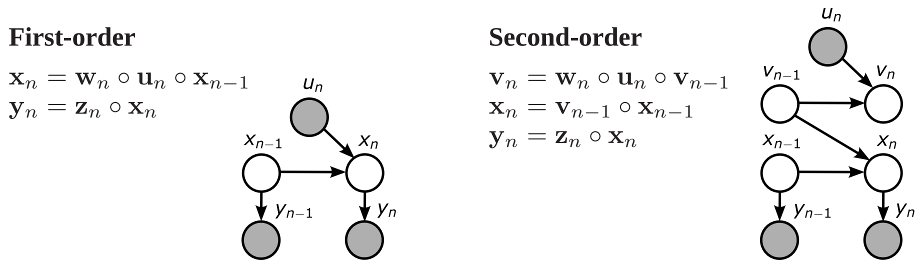


Figure 1: Process and graphical models for the discrete quaternion Bingham process.

rotation in 3-D space, the antipodal symmetry of the Bingham distribution correctly captures the topology of quaternion rotation space.

3 Discrete-time quaternion Bingham process

The first-order discrete-time quaternion Bingham process has, as its state, $\mathbf{x}_n$, a unit quaternion representing the orientation of interest at time n. The system's behavior is conditioned on control inputs $\mathbf{u}_n$, which are also unit quaternions. The new orientation is the old orientation rotated by the control input and then by independent noise $\mathbf{w_n} \sim \mathrm{Bingham}(\Lambda_p, V_p)$. Note that "$\circ$" denotes quaternion multiplication, which corresponds to composition of rotations for unit quaternions. ($\mathbf{q} \circ \mathbf{r}$ means "rotate by $\mathbf{r}$ and then by $\mathbf{q}$".)

The second-order quaternion Bingham process has state $(\mathbf{x}_n, \mathbf{v}_n)$, where $\mathbf{x}_n$ represents orientation and the quaternion $\mathbf{v}_n$ represents discrete rotational velocity at time n. The control inputs $\mathbf{u}_n$ are analogous to rotational accelerations. Process noise $\mathbf{w}_n$ enters the system in the velocity dynamics.

In both the first-order and second-order systems, observations $\mathbf{y}_n$ are given by the orientation $\mathbf{x}_n$ corrupted by independent Bingham noise $\mathbf{z_n} \sim \mathrm{Bingham}(\Lambda_o, V_o)$. One common choice for V_p and V_o is $\begin{bmatrix} 0&0&0 \\ 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{bmatrix}$, which means that the mode is the quaternion identity, $(1, 0, 0, 0)$. (Any V matrix whose top row contains all zeros will have this mode.) Figure 1 shows the process and graphical models for the discrete quaternion Bingham process.

4 Discrete quaternion Bingham filter

The state of a discrete-time quaternion Bingham process can be estimated using a discrete-time quaternion Bingham filter, which is a recursive estimator similar in structure to a Kalman filter. Unlike the Kalman filter, however, the QBF is approximate, in the sense that the tracked state distribution is projected to be in the Bingham family after every time step. The second-order QBF will also require an assumption of independence between $\mathbf{x}_n$ and $\mathbf{v}_n$, given all the data up to time n. Both the first-order and second-order QBFs are examples of *assumed density filtering*—a well-supported approximate inference method in the DBN literature [2]. We will start by deriving the first-order QBF, which follows the Kalman filter derivation quite closely. Note that the following derivations rely on several properties of Bingham distributions which will be detailed in section 5.

First-order QBF. Given a distribution over the initial state $\mathbf{x}_0$, $B_{\mathbf{x}_0} \sim \mathrm{Bingham}(\Lambda_0, V_0)$, and an action-observation sequence $\mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_n, \mathbf{y}_n$, the goal is to compute the posterior distribution $f(\mathbf{x}_n \mid \mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_n, \mathbf{y}_n)$. We can use Bayes' rule and the Markov property to decompose this distribution as follows:

$$\begin{aligned} B_{\mathbf{x}_n} &= f(\mathbf{x}_n|\mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_n, \mathbf{y}_n) \propto f(\mathbf{y}_n|\mathbf{x}_n) f(\mathbf{x}_n|\mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_{n-1}, \mathbf{y}_{n-1}, \mathbf{u}_n) \\ &= f(\mathbf{y}_n|\mathbf{x}_n) \int_{\mathbf{x}_{n-1}} f(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{u}_n) B_{\mathbf{x}_{n-1}}(\mathbf{x_{n-1}}) \\ &= f(\mathbf{y}_n|\mathbf{x}_n)(f_{\mathbf{w}_n} \circ \mathbf{u}_n \circ B_{\mathbf{x}_{n-1}})(\mathbf{x}_n) \quad . \end{aligned}$$

where $f_{\mathbf{w}_n} \circ \mathbf{u}_n \circ B_{\mathbf{x}_{n-1}}$ means rotate $B_{\mathbf{x}_{n-1}}$ by $\mathbf{u}_n$ and then convolve with the process noise distribution, $f_{\mathbf{w}_n}$. For the first term, $f(\mathbf{y}_n|\mathbf{x}_n)$, recall that the observation process is $\mathbf{y}_n = \mathbf{z}_n \circ \mathbf{x}_n$, so $\mathbf{y}_n|\mathbf{x}_n \sim Bingham(\mathbf{y_n}; \Lambda_o, V_o \circ \mathbf{x_n})$, where we used the result from section 5 for rotation of a Bingham by a fixed quaternion. Thus the distribution for $\mathbf{y}_n|\mathbf{x}_n$ is

$$f(\mathbf{y}_n|\mathbf{x}_n) = \frac{1}{F_o} \exp \sum_{i=1}^{3} \lambda_{oi} (\mathbf{y_n}^T (\mathbf{v_{oi}} \circ \mathbf{x_n}))^2 \enspace .$$

Now we can rewrite $\mathbf{y_n}^T(\mathbf{v_{oi}} \circ \mathbf{x_n})$ as $(\mathbf{v_{oi}^{-1}} \circ \mathbf{y_n})^T \mathbf{x_n}$, so that $f(\mathbf{y}_n|\mathbf{x}_n)$ is a Bingham density on $\mathbf{x_n}$, $\mathrm{Bingham}(\mathbf{x_n}; \Lambda_o, V_o^{-1} \circ \mathbf{y_n})$. Thus, computing $B_{\mathbf{x}_n}$ reduces to multiplying two Bingham PDFs on $\mathbf{x}_n$, which is given in section 5.

Second-order QBF. Given a factorized distribution over the initial state $f(\mathbf{x}_0, \mathbf{v}_0) = B_{\mathbf{x}_0} B_{\mathbf{v}_0}$ and an action-observation sequence $\mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_n, \mathbf{y}_n$, the goal is to compute the joint posterior distribution $f(\mathbf{x}_n, \mathbf{v}_n \mid \mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_n, \mathbf{y}_n)$. However, the joint distribution on $\mathbf{x}_n$ and $\mathbf{v}_n$ is too difficult to represent, so we instead compute the marginal posteriors over $\mathbf{x}_n$ and $\mathbf{v}_n$ separately, and approximate the joint posterior as the product of the marginals. The marginal posterior on $\mathbf{x}_n$ is

$$\begin{aligned}
B_{\mathbf{x}_n} &= f(\mathbf{x}_n|\mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_n, \mathbf{y}_n) \\
&\propto f(\mathbf{y}_n|\mathbf{x}_n) \int_{\mathbf{x}_{n-1}} f(\mathbf{x}_n|\mathbf{x}_{n-1}) B_{\mathbf{x}_{n-1}}(\mathbf{x}_{n-1}) \\
&= f(\mathbf{y}_n|\mathbf{x}_n)(B_{\mathbf{v}_{n-1}} \circ B_{\mathbf{x}_{n-1}})(\mathbf{x}_n)
\end{aligned}$$

since we assume $\mathbf{x}_{n-1}$ and $\mathbf{v}_{n-1}$ are independent given all the data up to time $n-1$.

Similarly, the marginal posterior on $\mathbf{v}_n$ is

$$\begin{aligned}
&B_{\mathbf{v}_n} = f(\mathbf{v}_n|\mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_n, \mathbf{y}_n) \\
&\quad \propto \int_{\mathbf{v}_{n-1}} f(\mathbf{v}_n \mid \mathbf{v}_{n-1}, \mathbf{u}_n) B_{\mathbf{v}_{n-1}}(\mathbf{v}_{n-1}) \int_{\mathbf{x}_n} f(\mathbf{y}_n|\mathbf{x}_n) f(\mathbf{x}_n|\mathbf{v}_{n-1}, \mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_{n-1}, \mathbf{y}_{n-1}).
\end{aligned}$$

Once again, $f(\mathbf{y}_n|\mathbf{x}_n)$ can be written as a Bingham density on $\mathbf{x}_n$, $\mathrm{Bingham}(\mathbf{x}_n; \Lambda_o, V_o^{-1} \circ \mathbf{y}_n)$. Next, note that $\mathbf{x}_n = \mathbf{v}_{n-1} \circ \mathbf{x}_{n-1}$ so that $f(\mathbf{x}_n|\mathbf{v}_{n-1}, \mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_{n-1}, \mathbf{y}_{n-1}) = B_{\mathbf{x}_{n-1}}(\mathbf{v}_{n-1}^{-1} \circ \mathbf{x}_n)$, which can also be re-written as a Bingham on $\mathbf{x}_n$. Now letting $\mathbf{x}_{n-1} \sim \mathrm{Bingham}(\Sigma, W)$, and since the product of two Bingham PDFs is Bingham, the integral over $\mathbf{x}_n$ becomes proportional to a Bingham normalization constant, ${}_1F_1(\frac{1}{2}; \frac{4}{2}; C(\mathbf{v}_{n-1}))$, where

$$C(\mathbf{v}_{n-1}) = \sum_{i=1}^{3} \Big(\sigma_i (\mathbf{v}_{n-1} \circ \mathbf{w}_i)(\mathbf{v}_{n-1} \circ \mathbf{w}_i)^T + \lambda_{oi} (\mathbf{v}_{oi}^{-1} \circ \mathbf{y}_n)(\mathbf{v}_{oi}^{-1} \circ \mathbf{y}_n)^T \Big) \enspace .$$

Comparing $C(\mathbf{v}_{n-1})$ with equation 4 in section 5 we find that ${}_1F_1(\frac{1}{2}; \frac{4}{2}; C(\mathbf{v}_{n-1})) \propto (f_{\mathbf{y}_n|\mathbf{x}_n} \circ B_{\mathbf{x}_{n-1}}^{-1})(\mathbf{v}_{n-1})$. Thus,

$$\begin{aligned}
B_{\mathbf{v}_n} &\propto \int_{\mathbf{v}_{n-1}} f(\mathbf{v}_n \mid \mathbf{v}_{n-1}, \mathbf{u}_n) B_{\mathbf{v}_{n-1}}(\mathbf{v}_{n-1}) \cdot (f_{\mathbf{y}_n|\mathbf{x}_n} \circ B_{\mathbf{x}_{n-1}}^{-1})(\mathbf{v}_{n-1}) \\
&= (f_{\mathbf{w}_n} \circ \mathbf{u}_n \circ (B_{\mathbf{v}_{n-1}} \cdot (f_{\mathbf{y}_n|\mathbf{x}_n} \circ B_{\mathbf{x}_{n-1}}^{-1})))(\mathbf{v}_n)
\end{aligned}$$

Note that $B_{\mathbf{v}_n}$ depends on $B_{\mathbf{x}_n}$, so $B_{\mathbf{x}_n}$ must be computed first.

5 Operations on Bingham distributions

In order to implement the quaternion Bingham filter, we need to be able to perform several operations on Bingham distributions. To our knowledge, all of these operations, except for computing the normalization constant, are new contributions of this paper. More operations (including calculation of KL-divergence and sampling methods) are presented in the appendix.

The Normalization constant. The primary difficulty with using the Bingham distribution in practice lies in computing the normalization constant, F. Since the distribution must integrate to one over its domain ($\mathbb{S}^d$), we can write the normalization constant as

$$F(\Lambda) = \int_{x \in \mathbb{S}^d} \exp\{\sum_{i=1}^{d} \lambda_i (\mathbf{v_i}^T \mathbf{x})^2\} = {}_1F_1(\frac{1}{2}; \frac{d+1}{2}; \Lambda) \tag{2}$$

where ${}_1F_1()$ is a hyper-geometric function of matrix argument [1]. Evaluating ${}_1F_1()$ is expensive, so we precompute a lookup table of F-values over a discrete grid of Λ's, and use tri-linear interpolation to quickly estimate normalizing constants on the fly.

Product of Bingham PDFs. The correction step of the filter requires multiplying PDFs. The product of two Bingham PDFs is given by adding their exponents:

$$\begin{aligned} f(x;\Lambda_1,V_1)f(x;\Lambda_2,V_2) &= \frac{1}{F_1F_2}\exp\{\mathbf{x}^T(\sum_{i=1}^{d}\lambda_{1i}\mathbf{v_{1i}}\mathbf{v_{1i}}^T+\lambda_{2i}\mathbf{v_{2i}}\mathbf{v_{2i}}^T)\mathbf{x}\} \\ &= \frac{1}{F_1F_2}\exp\{\mathbf{x}^T(C_1+C_2)\mathbf{x}\} \end{aligned} \tag{3}$$

After computing the sum $C = C_1 + C_2$ in the exponent of equation 3, we transform the PDF to standard form by computing the eigenvectors and eigenvalues of C, and then subtracting off the lowest magnitude eigenvalue from each spectral component, so that only the eigenvectors corresponding to the largest d eigenvalues (in magnitude) are kept, and $\lambda_1 \le \cdots \le \lambda_d \le 0$ (as in equation 6).

Rotation by a fixed quaternion. To find the effect of the control on the predictive distribution, we rotate the prior by $\mathbf{u}$. Given $\mathbf{q} \sim \mathrm{Bingham}(\Lambda, V)$, $\mathbf{u} \in \mathbb{S}^3$, and $\mathbf{s} = \mathbf{u}\circ\mathbf{q}$, then $\mathbf{s} \sim \mathrm{Bingham}(\Lambda, \mathbf{u}\circ V)$, where $\mathbf{u}\circ V \triangleq [\mathbf{u}\circ\mathbf{v_1}, \mathbf{u}\circ\mathbf{v_2}, \mathbf{u}\circ\mathbf{v_3}]$. In other words, $\mathbf{s}$ is distributed according to a Bingham whose orthogonal direction vectors have been rotated (on the left) by $\mathbf{u}$. Similarly, if $\mathbf{s} = \mathbf{q}\circ\mathbf{u}$ then $\mathbf{s} \sim \mathrm{Bingham}(\Lambda, V\circ\mathbf{u})$.

Proof for $\mathbf{s} = \mathbf{u}\circ\mathbf{q}$: Since unit quaternion rotation is invertible and volume-preserving, we have

$$f_{\mathbf{s}}(\mathbf{s}) = f_{\mathbf{q}}(\mathbf{u^{-1}}\circ\mathbf{s}) = \frac{1}{F}\exp\{\sum_{i=1}^{d}\lambda_i(\mathbf{v_i}^T(\mathbf{u^{-1}}\circ\mathbf{s}))^2\} = \frac{1}{F}\exp\{\sum_{i=1}^{d}\lambda_i((\mathbf{u}\circ\mathbf{v_i})^T\mathbf{s})^2\} \ .$$

Quaternion inversion. Given $\mathbf{q} \sim \mathrm{Bingham}(\Lambda, V)$ and $\mathbf{s} = \mathbf{q^{-1}}$, then $\mathbf{s} \sim \mathrm{Bingham}(\Lambda, JV)$, where J is the quaternion inverse matrix, $J = \begin{bmatrix} 1 & & & \\ & -1 & & \\ & & -1 & \\ & & & -1 \end{bmatrix}$. The proof follows the same logic as in the previous section.

Composition of quaternion Binghams. Implementing the QBF requires the computation of the PDF of the composition of two independent Bingham random variables. Letting $\mathbf{q} \sim \mathrm{Bingham}(\Lambda, V)$ and $\mathbf{r} \sim \mathrm{Bingham}(\Sigma, W)$, we wish to find the PDF of $\mathbf{s} = \mathbf{q}\circ\mathbf{r}$.

The **true distribution** is the convolution, in $\mathbb{S}^3$, of the PDFs of the component distributions[1].

$$\begin{aligned} f_{true}(\mathbf{s}) &= \int_{\mathbf{q}\in\mathbb{S}^3} f(\mathbf{s}|\mathbf{q})f(\mathbf{q}) = \frac{{}_1F_1(\frac{1}{2};\frac{4}{2};C(\mathbf{s}))}{2\cdot{}_1F_1(\frac{1}{2};\frac{4}{2};\Lambda){}_1F_1(\frac{1}{2};\frac{4}{2};\Sigma)} \\ \text{where} \qquad C(\mathbf{s}) &= \sum_{i=1}^{3}\Big(\sigma_i(\mathbf{s}\circ\mathbf{w_i}^{-1})(\mathbf{s}\circ\mathbf{w_i}^{-1})^T + \lambda_i\mathbf{v_i}\mathbf{v_i}^T\Big) \end{aligned} \tag{4}$$

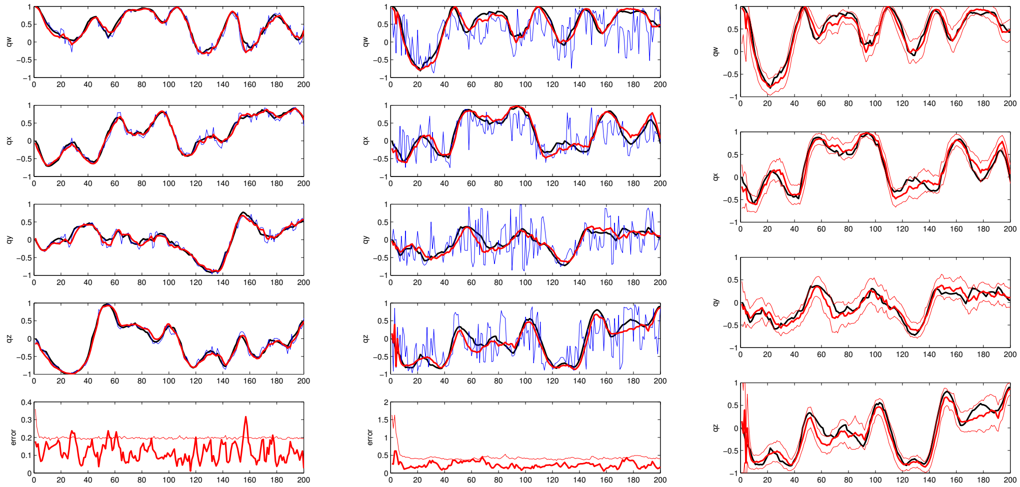
To **approximate** the PDF of $\mathbf{s}$ with a Bingham density, $f_B(\mathbf{s}) = f_B(\mathbf{q}\circ\mathbf{r})$, it is sufficient to compute the second moments of $\mathbf{q}\circ\mathbf{r}$, since the inertia matrix, $E[\mathbf{s}\mathbf{s}^T]$ is the sufficient statistic for the Bingham. This moment-matching approach is equivalent to the variational approach, where f_B is found by minimizing the KL divergence from f_B to f_{true}.

Noting that $\mathbf{q}\circ\mathbf{r}$ can be written as $(\mathbf{q}^T H_1^T\mathbf{r},\quad \mathbf{q}^T H_2^T\mathbf{r},\quad \mathbf{q}^T H_3^T\mathbf{r},\quad \mathbf{q}^T H_4^T\mathbf{r})$, where $H_1 = \begin{bmatrix} 1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1 \end{bmatrix}$, $H_2 = \begin{bmatrix} 0&-1&0&0\\1&0&0&0\\0&0&0&1\\0&0&-1&0 \end{bmatrix}$, $H_3 = \begin{bmatrix} 0&0&-1&0\\0&0&0&-1\\1&0&0&0\\0&1&0&0 \end{bmatrix}$, and $H_4 = \begin{bmatrix} 0&0&0&-1\\0&0&1&0\\0&-1&0&0\\1&0&0&0 \end{bmatrix}$, we find that

$$E[s_i s_j] = E[\mathbf{q}^T H_i^T\mathbf{r}\mathbf{r}^T H_j\mathbf{q}] \ ,$$

which has 16 terms of the form $\pm r_a r_b q_c q_d$, where $a,b,c,d \in \{1,2,3,4\}$. Since $\mathbf{q}$ and $\mathbf{r}$ are independent, $E[\pm r_a r_b q_c q_d] = \pm E[r_a r_b]E[q_c q_d]$. Therefore (by linearity of expectation), every entry in

[1] See the appendix for a full derivation.



(a) $\Lambda_o = (-50, -50, -50)$ (b) $\Lambda_o = (-10, -10, -1)$ (c) $\Lambda_o = (-10, -10, -1)$

Figure 2: Two simulated runs with the quaternion Bingham filter—(b) and (c) are different plots of the same simulation. In all figures, the thick black line is the true process signal, generated with isotropic process noise $\Lambda_p = (-400, -400, -400)$. The thin blue lines in (a) and (b) are the observation signal, and the thick red line is the filtered output. Rows 1-4 in each figure show the 4 quaternion dimensions (q_w, q_x, q_y, q_z). Row 5 in (a) and (b) shows the error between filter output and true state (thick red line), together with the QBF sample error 90% confidence bounds (thin red line). Marginal sample 90% confidence bounds are also shown in the thin red lines in (c).

the matrix $E[\mathbf{ss}^T]$ is a quadratic function of elements of the inertia matrices of $\mathbf{q}$ and $\mathbf{r}$, which can be easily computed given the Bingham normalization constants and their partial derivatives. The entire closed form for $E[\mathbf{ss}^T]$ is given in the appendix.

Entropy. The entropy of a Bingham distribution with PDF f is given by:

$$h(f) \;=\; -\int_{x\in\mathbb{S}^d} f(x)\log f(x) \;=\; \log F - \Lambda \cdot \frac{\nabla F}{F} \;. \tag{5}$$

The proof is given in the appendix. Since both the normalization constant, F, and its gradient with respect to Λ, ∇F, are stored in a lookup table, the entropy is trivial to approximate via interpolation, and can be used on the fly without any numerical integration over hyperspheres.

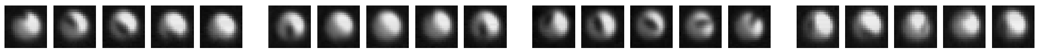
6 Experimental Results

We compare the quaternion Bingham filter against an extended Kalman filter (EKF) approach in quaternion space [13], where process and observation noise are generated by Gaussians in $\mathbb{R}^4$, the measurement function normalizes the quaternion state (to project it onto the unit hypersphere), and the state estimate is renormalized after every update. We chose the EKF both due to its popularity and because LaViola reports in [13] that it has similar (slightly better) accuracy to the unscented Kalman filter (UKF) in several real tracking experiments. We adapted two versions of the EKF (for first-order and second-order systems) from LaViola's EKF implementation by changing from a continuous to a discrete time prediction update. We also mapped QBF (Bingham) noise parameters to EKF (Gaussian) noise parameters by empirically matching second moments from the Bingham to the projected Gaussian—i.e., the Gaussian after it has been projected onto the unit hypersphere.

Synthetic Data. To test the first-order quaternion Bingham filter, we generated several synthetic signals by simulating a quaternion Bingham process, where the (velocity) controls were generated so that the nominal process state (before noise) would follow a sine wave pattern on each angle in Euler angle space. We chose this control pattern in order to cover a large area of 3-D rotation space with varying rotational velocities. Two examples of synthetic signals along with quaternion Bingham filter output are shown in figure 2. Their observation parameters were $\Lambda_o = (-50, -50, -50)$, which gives moderate, isotropic observation noise, and $\Lambda_o = (-10, -10, -1)$,

observation noise	-400, -400, -10			-400			-50		
process noise	-50	-200	-400	-50	-200	-400	-50	-200	-400
% improvement	**37.3**	**45.6**	**54.3**	**19.9**	**3.33**	**1.52**	**3.42**	**0.72**	**0.47**
$\pm$	(5.1)	(5.0)	(6.4)	(1.8)	(0.53)	(0.44)	(0.88)	(0.40)	(0.27)

Table 1: Projected Gaussian process simulations. Average % mean error decrease for QBF over EKF.



(a) slow top-spin (b) fast top-spin (c) slow side-spin (d) fast side-spin

Figure 3: Example image sequences from the spinning ping-pong ball dataset. In addition to lighting variations and low image resolution, high spin rates make this dataset extremely challenging for orientation tracking algorithms. Also, because the cameras were facing top-down towards the table, tracking side-spin relies on correctly estimating the orientation of the elliptical marking in the image, and is therefore much harder than tracking top-spin or under-spin.

which yields moderately high noise in the first two directions, and near-uniform noise in the third direction. We estimated the composition approximation error (KL-divergence) for 9 of these signals, with both isotropic and nonisotropic noise models, from all combinations of (Λ_p, Λ_o) in $\{(-50, -50, -50), (-200, -200, -200), (-10, -10, -1)\}$. The mean composition error was .0012, while the max was .0197, which occurred when Λ_p and Λ_o were both $(-10, -10, -1)$.

For the EKF comparison, we wanted to give the EKF the best chance to succeed, so we generated the data from a projected Gaussian process, with process and observation noise generated according to a projected Gaussian (in order to match the EKF dynamics model) rather than from Bingham distributions. We ran the first-order QBF and EKF on 270 synthetic projected Gaussian process signals (each with 1000 time steps) with different amounts of process and observation noise, and found the QBF to be more accurate than the EKF on 268/270 trials. The mean angular change in 3-D orientation between time steps were 7, 9, and 18 degrees for process noise parameters -400, -200, and -50, respectively (where -400 means $\Lambda_p = (-400, -400 - 400)$, etc.).

The most extreme cases involved anisotropic observation noise, with an average improvement over the EKF mean error rate of 40-50%. The combination of high process noise and low observation noise also causes trouble for the EKF. Table 1 summarizes the results.

Spinning ping-pong ball dataset To test the second-order QBF, we collected a dataset of high-speed videos of 73 spinning ping-pong balls in flight[2] (Figure 3). On each ball we drew a solid black ellipse over the ball's logo to allow the high-speed (200fps) vision system to estimate the ball's orientation by finding the position and orientation of the logo. However, an ellipse was only drawn on one side of each ball, so the ball's orientation could only be estimated when the logo was visible in the image. Also, since ellipses are symmetric, each logo detection has two possible orientation interpretations[3]. The balls were spinning at 25-50 revolutions per second (which equates to a 45-90 degree orientation change per frame), making the filtering problem extremely challenging due to aliasing effects. We used a ball gun to shoot the balls with consistent spin and speed, at 4 different spin settings (topspin, underspin, left-sidespin, and right-sidespin) and 3 different speed settings (slow, medium, fast), for a total of 12 different spin types. Although we initially collected videos of 107 ball trajectories, the logo could only be reliably found in 73 of them; the remaining 34 videos were discarded. Although not our current focus, adding more markings to the ball and improving logo detections would allow the ball's orientation and spin to be tracked on a larger percentage of such videos. To establish an estimate of ground truth, we then manually labeled each ball image with the position and orientation of the logo (when visible), from which we recovered the ball orientation (up to symmetry). We

[2] In future work, we plan to incorporate control signals into the ping-pong spin tracking dataset by allowing the ball to bounce during the trajectory.

[3] We disambiguated between the two possible ball orientation observations by picking the observation with highest likelihood under the current QBF belief.

then used least-squares non-linear regression to smooth out our (noisy) manual labels by finding the constant rotation, $\hat{\mathbf{s}}$, which best fit the labeled orientations for each trajectory[4].

To run the second-order QBF on this data, we initialized the QBF with a uniform orientation distribution $B_{\mathbf{x}_0}$ and a low concentration ($\Lambda = (-3, -3 - 3)$) spin distribution $B_{\mathbf{v}_0}$ centered on the identity rotation, $(1, 0, 0, 0)$. In other words, we provided no information about the ball's initial orientation, and an extremely weak bias towards slower spins. We also treated the "no-logo-found" (a.k.a. "dark side") observations as a very noisy observation of the logo in the center of the back side of the ball at an arbitrary orientation, with $\Lambda_o = (-3.6, -3.6, 0)$[5]. When the logo was detected, we used $\Lambda_o = (-40, -40, -10)$ for the observation noise. A process noise with $\Lambda_p = (-400, -400, -400)$ was used throughout, to account for small perturbations to spin.

Results of running the second-order QBF (QBF-2) are shown in figure 4. We compared the second-order QBF to the second-order EKF (EKF-2) and also to the first-order QBF and EKF (QBF-1 and EKF-1), which were given the difference between subsequent orientation observations as their observations of spin. The solid, thin, blue line in each plot marked "oracle prior" shows results from running QBF-2 with a best-case-scenario prior, centered on the average ground truth spin for that spin type, with $\Lambda = (-10, -10, -10)$. We show mean orientation and spin errors (to regressed ground truth), and also spin classification accuracy using the MAP estimate of spin type (out of 12) given the current spin belief[6]. The results clearly show that QBF-2 does the best job of identifying and tracking the ball rotations on this extremely challenging dataset, achieving a classification rate of 91% after just 30 video frames, and a mean spin (quaternion) error of 0.17 radians (10 degrees), with an average of 6.1 degrees of logo axis error and 6.8 degrees of logo angle error. In contrast, the EKF-2 does not significantly outperform random guessing, due to the extremely large observation noise and spin rates in this dataset. In the middle of the pack are QBF-1 and EKF-1, which converge much more slowly since they use the raw observations (rather than the smoothed orientation signal used by QBF-2) to estimate ball spin. Finally, to address the aliasing problem, we ran a set of 12 QBFs in parallel, each with a different spin prior mode (one for each spin type), with $\Lambda = (-10, -10, -10)$. At each time step, the filter was selected with the highest total data likelihood. Results of this "filter bank" approach are shown in the solid, thin, green line in figure 4.

7 Conclusion

For many control and vision applications, the state of a dynamic process involving 3-D orientations and spins must be estimated over time, given noisy observations. Previously, such estimation was limited to slow-moving signals with low-noise observations, where linear approximations to 3-D rotation space were adequate. The contribution of our approach is that the quaternion Bingham filter encodes uncertainty directly on the unit quaternion hypersphere, using a distribution—the Bingham—with nice mathematical properties enabling efficient approximate inference, with no restrictions on the magnitude of process dynamics or observation noise. Because of the compact nature of 3-D rotation space and the flexibility of the Bingham distribution, we can use the QBF not only for tracking but also for identification of signals, by starting the QBF with an extremely unbiased prior, a feat which previously could only be matched by more computationally-intensive algorithms, such as discrete Bayesian filters or particle filters. The one drawback of the quaternion Bingham process is that its process and observation models are somewhat restricted; for example, no quaternion exponentiation (or interpolation) is currently allowed, which would be required for a continuous-time update. Extending the QBF for these cases (most likely using numerical methods) is a direction for future work.

[4]Due to air resistance and random perturbations, the spin was not really constant throughout each trajectory. But for the short duration of our experiments (40 frames), the constant spin approximation was sufficient.

[5]We got this Λ_o by fitting a Bingham to all possible dark side orientations.

[6]Spin was classified into one of the 12 spin types by taking the average ground truth spin for each spin type and choosing the one with the highest likelihood with respect to the current spin belief.

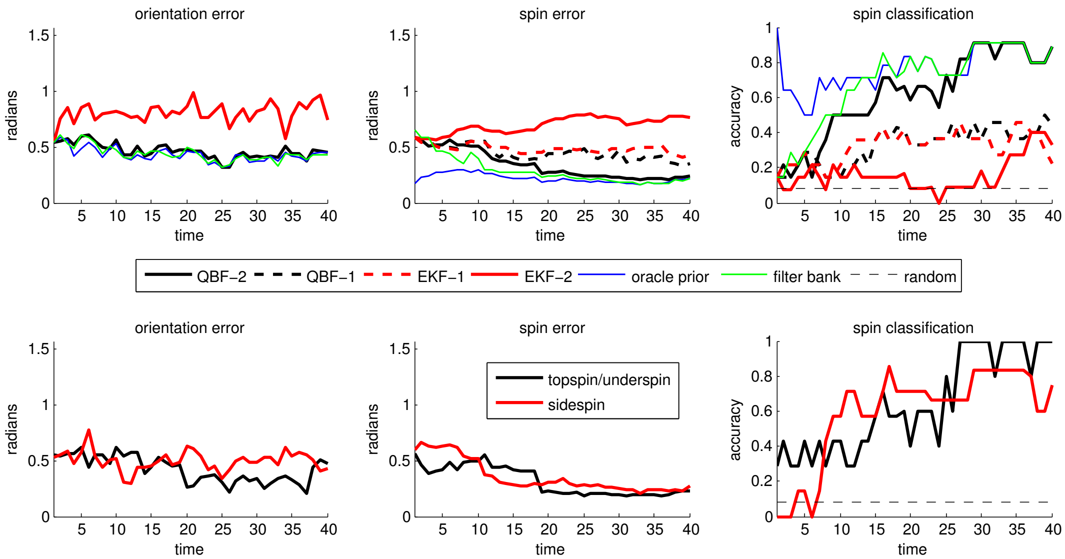


Figure 4: Spinning ping-pong ball tracking results. Top row: comparison of QBF-2 (with and without an oracle-given prior) to QBF-1, EKF-1, EKF-2, and random guessing (for spin classification); QBF-1 and EKF-1 do not show up in the orientation error graph because they only tracked spin. Note that QBF-2 quickly converges to the oracle error and classification rates. Bottom row: QBF-2 results broken down into top-spin/under-spin vs. side-spin. As mentioned earlier, the side-spin data is harder to track due to the chosen camera placement and ball markings for this experiment.

A Formulas and Derivations for the Bingham Distribution

A.1 Bingham Distribution

To review, the probability density function (PDF) for the Bingham distribution is

$$f(\mathbf{x};\Lambda,V) = \frac{1}{F}\exp\{\sum_{i=1}^{d}\lambda_i(\mathbf{v_i}^T\mathbf{x})^2\} \tag{6}$$

where $\mathbf{x}$ is a unit vector on the surface of the sphere $\mathbb{S}^d \subset \mathbb{R}^{d+1}$, F is a normalization constant, Λ is a vector of concentration parameters, and the columns of the $(d+1)\times d$ matrix V are orthogonal unit vectors. We refer to the density in equation 6 as the *standard form* for the Bingham. A more succinct form for the Bingham is $f(\mathbf{x};C) = \frac{1}{F}\exp(\mathbf{x}^T C\mathbf{x})$, where C is a $(d+1)\times(d+1)$ orthogonal matrix; however, C is clearly an over-parameterization since the first d columns of C completely determine its final column. By convention, one typically defines Λ and V so that $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_d \le 0$. Note that a large-magnitude λ_i indicates that the distribution is highly peaked along the direction $\mathbf{v_i}$, while a small-magnitude λ_i indicates that the distribution is spread out along $\mathbf{v_i}$.

A.2 The Normalization constant

The primary difficulty with using the Bingham distribution in practice lies in computing the normalization constant, F. Since the distribution must integrate to one over its domain ($\mathbb{S}^d$), we can write the normalization constant as

$$F(\Lambda) = \int_{x\in\mathbb{S}^d}\exp\{\sum_{i=1}^{d}\lambda_i(\mathbf{v_i}^T\mathbf{x})^2\} \tag{7}$$

In general, there is no closed form for this integral, which means that F must be approximated. Typically, this is done via series expansion [1, 8], although saddle-point approximations [11] have also been used. Following Bingham [1], we note that $F(\Lambda)$ is proportional to a hyper-geometric function of matrix argument, with series expansion

$$
\begin{aligned}
F(\Lambda) &= 2 \cdot {}_1F_1(\frac{1}{2}; \frac{d+1}{2}; \Lambda) \\
&= 2\sqrt{\pi} \sum_{\alpha_1=0}^{\infty} \cdots \sum_{\alpha_d=0}^{\infty} \frac{\Gamma(\alpha_1 + \frac{1}{2}) \cdots \Gamma(\alpha_d + \frac{1}{2})}{\Gamma(\alpha_1 + \cdots + \alpha_d + \frac{d+1}{2})} \cdot \frac{\lambda_1^{\alpha_1} \cdots \lambda_d^{\alpha_d}}{\alpha_1! \cdots \alpha_d!}
\end{aligned} \tag{8}
$$

For practical usage, we precompute a lookup table of F-values over a discrete grid of Λ's, and use interpolation to quickly estimate normalizing constants on the fly.

A.3 Max Likelihood Parameter Estimation

Following Bingham [1], we estimate the parameters V and Λ given a set of N samples, $\{\mathbf{x_i}\}$, using a maximum likelihood approach. Finding the maximum likelihood estimate (MLE) $\hat{V}$ is an eigenvalue problem—the MLE mode of the distribution is equal to the eigenvector of the inertia matrix $S = \frac{1}{N}\sum_i \mathbf{x_i}\mathbf{x_i^T}$ corresponding to the largest eigenvalue, while the columns of $\hat{V}$ are equal to the eigenvectors corresponding to the 2nd through $(d+1)$th eigenvalues of S.

The maximum likelihood estimate $\hat{\Lambda}$ is found by setting the partial derivatives of the data log likelihood function with respect to Λ to zero, yielding

$$
\frac{1}{F(\Lambda)} \frac{\partial F(\Lambda)}{\partial \lambda_j} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{v_j}^T \mathbf{x_i})^2 = \mathbf{v_j}^T S \mathbf{v_j}, \tag{9}
$$

for $j = 1, \ldots, d$. Just as we did for $F(\Lambda)$, we can pre-compute values of the gradient of F with respect to Λ, ∇F, and store them in a lookup table. Using a kD-tree, we can find the nearest neighbors of a new sample $\nabla F / F$ in $O(d \log M)$ time (where M^d is the size of the lookup table), and use their indices to find Λ via interpolation (since the lookup tables for F and ∇F are indexed by Λ).

Notice that the maximum likelihood estimates for V and Λ are both computed given only the inertia matrix, S. Thus, S is a sufficient statistic for the Bingham distribution. In fact, there is a beautiful result from the theory of exponential families which says that the Bingham distribution is the *maximum entropy* distribution on the hypersphere which matches the sample inertia matrix, $S = E[\mathbf{x}\mathbf{x}^T]$ [14]. This gives us further theoretical justification to use the Bingham distribution in practice, if we assume that all of the relevant information about the data is captured in the inertia matrix.

A.4 Mode

The mode of the Bingham, $\boldsymbol{\mu}$, is the unit vector which is orthogonal to $\mathbf{v_1}, \ldots, \mathbf{v_d}$.

A.5 Inertia Matrix

Equation 9 can be inverted to solve for the inertia matrix, S, of a Bingham distribution. The eigenvectors of S are given by $\mathbf{v_1}, \ldots, \mathbf{v_d}$, together with the mode of the Bingham, $\boldsymbol{\mu}$. The first d eigenvalues are given by $\frac{\nabla F}{F}$, while the last eigenvalue is $1 - \sum_i \frac{\partial F / \partial \lambda_i}{F}$.

A.6 Sampling

There are many situations in which it is important to be able to draw samples from a distribution. In this paper, we used sampling to compute empirical confidence bounds on the predictions of the quaternion Bingham filter. In addition, one might use a Bingham sampler in the construction of a particle filter for a quaternion Bingham process.

Because of the complexity of the normalization constant, sampling from the Bingham distribution directly is difficult. Therefore, we use a Metropolis-Hastings sampler, with target distribution given by the Bingham density, and proposal distribution given by the projected zero-mean Gaussian[7] in

[7] To sample from the projected Gaussian, we first sample from a Gaussian with covariance S, then project the sample onto the unit sphere.

$\mathbb{R}^{d+1}$ with covariance matrix equal to the Bingham's sample inertia matrix, S. Because the proposal distribution is very similar to the target distribution, the Metropolis-Hastings sampler typically has a very high acceptance rate, making it an efficient sampling method.

A.7 Entropy

The entropy of a Bingham distribution with PDF f is given by:

$$\begin{aligned} h(f) &= -\int_{x\in\mathbb{S}^d} f(x)\log f(x) \\ &= -\int_{x\in\mathbb{S}^d} \frac{1}{F} e^{\mathbf{x}^T C\mathbf{x}} (\mathbf{x}^T C\mathbf{x} - \log F) \\ &= \log F - \frac{1}{F}\int_{x\in\mathbb{S}^d} \mathbf{x}^T C\mathbf{x} e^{\mathbf{x}^T C\mathbf{x}}. \end{aligned}$$

Writing f in standard form, and denoting the hyperspherical integral $\int_{x\in\mathbb{S}^d} \mathbf{x}^T C\mathbf{x} e^{\mathbf{x}^T C\mathbf{x}}$ by $g(\Lambda)$,

$$\begin{aligned} g(\Lambda) &= \int_{x\in\mathbb{S}^d} \sum_{i=1}^{d} \lambda_i (\mathbf{v_i}^T\mathbf{x})^2 e^{\sum_{j=1}^{d}\lambda_j(\mathbf{v_j}^T\mathbf{x})^2} \\ &= \sum_{i=1}^{d} \lambda_i \frac{\partial F}{\partial \lambda_i} = \Lambda\cdot\nabla F. \end{aligned}$$

Thus, the entropy is

$$h(f) \;=\; -\int_{x\in\mathbb{S}^d} f(x)\log f(x) \;=\; \log F - \Lambda\cdot\frac{\nabla F}{F}\;. \tag{10}$$

Since both the normalization constant, F, and its gradient with respect to Λ, ∇F, are stored in a lookup table, the entropy is trivial to compute, and can be used on the fly without any numerical integration over hyperspheres.

A.8 Divergence

The Kullback-Leibler (KL) divergence between two Binghams with PDFs f and g is

$$D_{KL}(f \parallel g) = -\int_{\mathbf{x}\in\mathbb{S}^d} f(\mathbf{x})\log\frac{f(\mathbf{x})}{g(\mathbf{x})} \tag{11}$$

$$= h(f,g) - h(f), \tag{12}$$

where $h(f)$ is the entropy of f and $h(f,g)$ is the cross entropy of f and g. Using similar methods as in the entropy derivation, $h(f,g)$ is given by

$$h(f,g) = \log F_g - \sum_{i=1}^{d}\lambda_{fi}\left(w_{i0}^2 + \sum_{j=1}^{d}(w_{ij}^2 - w_{i0}^2)\frac{1}{F_f}\frac{\partial F_f}{\partial\lambda_{fj}}\right), \tag{13}$$

where the matrix W of coefficients w_{ij} is given by the rotation of g's direction vectors, V_g, into f's full-rank basis, $V_f' = [\boldsymbol{\mu_f} \quad V_f]$ (where $\boldsymbol{\mu_f}$ is the mode of f),

$$W = (V_f')^{-1}V_g. \tag{14}$$

Thus, the cost of a KL divergence is merely a $d\times d$ matrix inversion, plus the time it takes to look up the normalization constants and their gradients.

A.9 Composition

Letting $\mathbf{q}\sim\mathrm{Bingham}(\Lambda, V)$ and $\mathbf{r}\sim\mathrm{Bingham}(\Sigma, W)$, we wish to find the PDF of $\mathbf{s} = \mathbf{q}\circ\mathbf{r}$, where "$\circ$" denotes quaternion multiplication, which corresponds to composition of rotations for unit quaternions.

A.9.1 True Distribution

The **true distribution** is the convolution, in $\mathbb{S}^3$, of the PDFs of the component distributions.

$$
\begin{aligned}
f(\mathbf{s}) &= \int_{\mathbf{q}\in\mathbb{S}^3} f(\mathbf{s}|\mathbf{q})f(\mathbf{q}) = \int_{\mathbf{q}} f(\mathbf{s};\Sigma,\mathbf{q}\circ W)f(\mathbf{q};\Lambda,V) \\
&= \frac{1}{F(\Sigma)F(\Lambda)}\int_{\mathbf{q}} \exp\sum_{i=1}^{3}\left[\sigma_i((\mathbf{s}\circ\mathbf{w_i}^{-1})^T\mathbf{q})^2 + \lambda_i(\mathbf{v_i}^T\mathbf{q})^2\right] \\
&= \frac{1}{F(\Sigma)F(\Lambda)}\int_{\mathbf{q}} e^{\mathbf{q}^T C(\mathbf{s})\mathbf{q}}
\end{aligned}
$$

where $\qquad C(\mathbf{s}) = \sum_{i=1}^{3}\left(\sigma_i(\mathbf{s}\circ\mathbf{w_i}^{-1})(\mathbf{s}\circ\mathbf{w_i}^{-1})^T + \lambda_i\mathbf{v_i}\mathbf{v_i}^T\right)$.

Since all the λ_i's and σ_i's are negative, the matrix C is negative semidefinite. Therefore, $\int_{\mathbf{q}} e^{\mathbf{q}^T C(\mathbf{s})\mathbf{q}}$ is the normalization constant of a Bingham distribution, so that the final form for the composed PDF is a quotient of hypergeometric functions,

$$
f_{true}(\mathbf{s}) = \frac{{}_1F_1(\frac{1}{2};\frac{4}{2};C(\mathbf{s}))}{2\cdot{}_1F_1(\frac{1}{2};\frac{4}{2};\Lambda){}_1F_1(\frac{1}{2};\frac{4}{2};\Sigma)} \quad . \tag{15}
$$

A.9.2 Method of Moments: Inertia Matrix

Let $A = [a_{ij}] = E[\mathbf{q}\mathbf{q}^T]$ and $B = [b_{ij}] = E[\mathbf{r}\mathbf{r}^T]$ be the inertia matrices of $\mathbf{q}$ and $\mathbf{r}$, respectively. Then $E[\mathbf{s}\mathbf{s}^T]$ is given by:

$$
\begin{bmatrix}
\begin{array}{r} a_{11}b_{11} - 2a_{12}b_{12} \\ -2a_{13}b_{13} - 2a_{14}b_{14} \\ +a_{22}b_{22} + 2a_{23}b_{23} \\ +2a_{24}b_{24} + a_{33}b_{33} \\ +2a_{34}b_{34} + a_{44}b_{44} \end{array} &
\begin{array}{r} a_{11}b_{12} + a_{12}b_{11} + a_{13}b_{14} \\ -a_{14}b_{13} - a_{12}b_{22} - a_{22}b_{12} \\ -a_{13}b_{23} - a_{23}b_{13} - a_{14}b_{24} \\ -a_{24}b_{14} - a_{23}b_{24} + a_{24}b_{23} \\ -a_{33}b_{34} + a_{34}b_{33} - a_{34}b_{44} \\ +a_{44}b_{34} \end{array} &
\begin{array}{r} a_{11}b_{13} + a_{13}b_{11} - a_{12}b_{14} \\ +a_{14}b_{12} - a_{12}b_{23} - a_{23}b_{12} \\ -a_{13}b_{33} + a_{22}b_{24} - a_{24}b_{22} \\ -a_{33}b_{13} - a_{14}b_{34} - a_{34}b_{14} \\ +a_{23}b_{34} - a_{34}b_{23} + a_{24}b_{44} \\ -a_{44}b_{24} \end{array} &
\begin{array}{r} a_{11}b_{14} + a_{12}b_{13} - a_{13}b_{12} \\ +a_{14}b_{11} - a_{12}b_{24} - a_{24}b_{12} \\ -a_{22}b_{23} + a_{23}b_{22} - a_{13}b_{34} \\ -a_{34}b_{13} - a_{23}b_{33} + a_{33}b_{23} \\ -a_{14}b_{44} - a_{24}b_{34} + a_{34}b_{24} \\ -a_{44}b_{14} \end{array} \\
\begin{array}{r} a_{11}b_{12} + a_{12}b_{11} + a_{13}b_{14} \\ -a_{14}b_{13} - a_{12}b_{22} - a_{22}b_{12} \\ -a_{13}b_{23} - a_{23}b_{13} - a_{14}b_{24} \\ -a_{24}b_{14} - a_{23}b_{24} + a_{24}b_{23} \\ -a_{33}b_{34} + a_{34}b_{33} - a_{34}b_{44} \\ +a_{44}b_{34} \end{array} &
\begin{array}{r} 2a_{12}b_{12} + a_{11}b_{22} \\ +a_{22}b_{11} + 2a_{13}b_{24} \\ -2a_{14}b_{23} + 2a_{23}b_{14} \\ -2a_{24}b_{13} - 2a_{34}b_{34} \\ +a_{33}b_{44} + a_{44}b_{33} \end{array} &
\begin{array}{r} a_{12}b_{13} + a_{13}b_{12} + a_{11}b_{23} \\ +a_{23}b_{11} - a_{12}b_{24} + a_{14}b_{22} \\ -a_{22}b_{14} + a_{24}b_{12} + a_{13}b_{34} \\ -a_{14}b_{33} + a_{33}b_{14} - a_{34}b_{13} \\ +a_{24}b_{34} + a_{34}b_{24} - a_{23}b_{44} \\ -a_{44}b_{23} \end{array} &
\begin{array}{r} a_{12}b_{14} + a_{14}b_{12} + a_{11}b_{24} \\ +a_{12}b_{23} - a_{13}b_{22} + a_{22}b_{13} \\ -a_{23}b_{12} + a_{24}b_{11} - a_{14}b_{34} \\ +a_{34}b_{14} + a_{13}b_{44} + a_{23}b_{34} \\ -a_{24}b_{33} - a_{33}b_{24} + a_{34}b_{23} \\ -a_{44}b_{13} \end{array} \\
\begin{array}{r} a_{11}b_{13} + a_{13}b_{11} - a_{12}b_{14} \\ +a_{14}b_{12} - a_{12}b_{23} - a_{23}b_{12} \\ -a_{13}b_{33} + a_{22}b_{24} - a_{24}b_{22} \\ -a_{33}b_{13} - a_{14}b_{34} - a_{34}b_{14} \\ +a_{23}b_{34} - a_{34}b_{23} + a_{24}b_{44} \\ -a_{44}b_{24} \end{array} &
\begin{array}{r} a_{12}b_{13} + a_{13}b_{12} + a_{11}b_{23} \\ +a_{23}b_{11} - a_{12}b_{24} + a_{14}b_{22} \\ -a_{22}b_{14} + a_{24}b_{12} + a_{13}b_{34} \\ -a_{14}b_{33} + a_{33}b_{14} - a_{34}b_{13} \\ +a_{24}b_{34} + a_{34}b_{24} - a_{23}b_{44} \\ -a_{44}b_{23} \end{array} &
\begin{array}{r} 2a_{13}b_{13} + 2a_{14}b_{23} \\ -2a_{23}b_{14} + a_{11}b_{33} \\ +a_{33}b_{11} - 2a_{12}b_{34} \\ +2a_{34}b_{12} - 2a_{24}b_{24} \\ +a_{22}b_{44} + a_{44}b_{22} \end{array} &
\begin{array}{r} a_{13}b_{14} + a_{14}b_{13} - a_{13}b_{23} \\ +a_{23}b_{13} + a_{14}b_{24} - a_{24}b_{14} \\ +a_{11}b_{34} + a_{12}b_{33} - a_{33}b_{12} \\ +a_{34}b_{11} + a_{23}b_{24} + a_{24}b_{23} \\ -a_{12}b_{44} - a_{22}b_{34} - a_{34}b_{22} \\ +a_{44}b_{12} \end{array} \\
\begin{array}{r} a_{11}b_{14} + a_{12}b_{13} - a_{13}b_{12} \\ +a_{14}b_{11} - a_{12}b_{24} - a_{24}b_{12} \\ -a_{22}b_{23} + a_{23}b_{22} - a_{13}b_{34} \\ -a_{34}b_{13} - a_{23}b_{33} + a_{33}b_{23} \\ -a_{14}b_{44} - a_{24}b_{34} + a_{34}b_{24} \\ -a_{44}b_{14} \end{array} &
\begin{array}{r} a_{12}b_{14} + a_{14}b_{12} + a_{11}b_{24} \\ +a_{12}b_{23} - a_{13}b_{22} + a_{22}b_{13} \\ -a_{23}b_{12} + a_{24}b_{11} - a_{14}b_{34} \\ +a_{34}b_{14} + a_{13}b_{44} + a_{23}b_{34} \\ -a_{24}b_{33} - a_{33}b_{24} + a_{34}b_{23} \\ -a_{44}b_{13} \end{array} &
\begin{array}{r} a_{13}b_{14} + a_{14}b_{13} - a_{13}b_{23} \\ +a_{23}b_{13} + a_{14}b_{24} - a_{24}b_{14} \\ +a_{11}b_{34} + a_{12}b_{33} - a_{33}b_{12} \\ +a_{34}b_{11} + a_{23}b_{24} + a_{24}b_{23} \\ -a_{12}b_{44} - a_{22}b_{34} - a_{34}b_{22} \\ +a_{44}b_{12} \end{array} &
\begin{array}{r} 2a_{14}b_{14} - 2a_{13}b_{24} \\ +2a_{24}b_{13} + 2a_{12}b_{34} \\ -2a_{23}b_{23} - 2a_{34}b_{12} \\ +a_{11}b_{44} + a_{22}b_{33} \\ +a_{33}b_{22} + a_{44}b_{11} \end{array}
\end{bmatrix}
$$

A.9.3 Estimating the error of approximation

To estimate the error in the Bingham approximation to the composition of two quaternion Bingham distributions, $B_1 \circ B_2$, we approximate the KL divergence from f_B to f_{true} using a finite element approximation on the quaternion hypersphere

$$
D_{KL}(f_B \parallel f_{true}) = -\int_{\mathbf{x}\in\mathbb{S}^3} f_B(\mathbf{x})\log\frac{f_B(\mathbf{x})}{f_{true}(\mathbf{x})} \approx \sum_{\mathbf{x}\in\mathcal{F}(\mathbb{S}^3)} f_B(\mathbf{x})\log\frac{f_B(\mathbf{x})}{f_{true}(\mathbf{x})}\cdot\Delta\mathbf{x}
$$

where $\mathcal{F}(\mathbb{S}^d)$ and $\{\Delta\mathbf{x}\}$ are the points and volumes of the finite-element approximation to $\mathbb{S}^3$, based on a recursive tetrahedral-octahedral subdivision method [17].

References

[1] Christopher Bingham. An antipodally symmetric distribution on the sphere. *The Annals of Statistics*, 2(6):1201–1225, November 1974.

[2] Xavier Boyen and Daphne Koller. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, page 3342, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[3] Yee-Jin Cheon and Jong-Hwan Kim. Unscented filtering in a unit quaternion space for spacecraft attitude estimation. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, pages 66–71, 2007.

[4] D. Choukroun, I.Y. Bar-Itzhack, and Y. Oshman. Novel quaternion Kalman filter. *Aerospace and Electronic Systems, IEEE Transactions on*, 42(1):174–190, 2006.

[5] I. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, 1998.

[6] Wendelin Feiten, Pradeep Atwal, Robert Eidenberger, and Thilo Grundmann. 6D pose uncertainty in robotic perception. In *Advances in Robotics Research*, pages 89–98. Springer Berlin Heidelberg, 2009.

[7] Jared Glover, Radu Rusu, and Gary Bradski. Monte carlo pose estimation with quaternion kernels and the bingham distribution. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

[8] John T. Kent. Asymptotic expansions for the Bingham distribution. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 36(2):139–144, 1987.

[9] A. Kim and M.F. Golnaraghi. A quaternion-based orientation estimation algorithm using an inertial measurement unit. In *Position Location and Navigation Symposium, 2004. PLANS 2004*, pages 268–272, 2004.

[10] E. Kraft. A quaternion-based unscented Kalman filter for orientation tracking. In *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, volume 1, pages 47–54, 2003.

[11] A. Kume and Andrew T. A. Wood. Saddlepoint approximations for the Bingham and Fisher-Bingham normalising constants. *Biometrika*, 92(2):465–476, June 2005.

[12] Karsten Kunze and Helmut Schaeben. The Bingham distribution of quaternions and its spherical radon transform in texture analysis. *Mathematical Geology*, 36(8):917–943, November 2004.

[13] J.J. LaViola. A comparison of unscented and extended Kalman filtering for estimating quaternion motion. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2435–2440 vol.3, 2003.

[14] K. V. Mardia. Characterizations of directional distributions. In *Statistical Distributions in Scientific Work*, volume 3, pages 365–385. D. Reidel Publishing Company, Dordrecht, 1975.

[15] J.L. Marins, Xiaoping Yun, E.R. Bachmann, R.B. McGhee, and M.J. Zyda. An extended Kalman filter for quaternion-based orientation estimation using MARG sensors. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, pages 2003–2011 vol.4, 2001.

[16] Stephen R Niezgoda and Jared Glover. Unsupervised learning for efficient texture estimation from limited discrete orientation data. *Metallurgical and Materials Transactions A*, pages 1–15, 2013.

[17] S. Schaefer, J. Hakenberg, and J. Warren. Smooth subdivision of tetrahedral meshes. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 147–154, Nice, France, 2004. ACM.