A Geodesic-Aware Quaternion Neural Network for Orientation Estimation

Oumar Butt*, Alin Tisan*, Danilo Mandic†, Clive Cheong Took*
*Electronic Engineering, Royal Holloway University of London
†Electrical and Electronic Engineering, Imperial College London
{oumar.butt, clive.cheongtook}@rhul.ac.uk, alin.tisan@ieee.org, d.mandic@ic.ac.uk

***Abstract*—Quaternion-valued neural networks are the natural candidates for orientation estimation due to their numerous advantages in modelling rotations. However, existing quaternion algorithms ignore the curved geometry of the hypersphere $\mathbb{S}^3$. Perhaps, this is because most quaternion neural networks have been designed for general multi-dimensional processing, making the implicit assumption of Euclidean space. To address this shortcoming, this paper proposes a quaternion neural network with a geodesic loss that accounts for the curved structure in $\mathbb{S}^3$. More importantly, our loss formulation explicitly accounts for the ambiguous equivalence $q \equiv -q$ in quaternion rotation. Simulation studies on both synthetic and the real-world aerial vehicle dataset demonstrate an order-of-magnitude reduction in mean angular tracking error compared to Euclidean methods.**

***Index Terms*—quaternions, neural networks, geodesic distance, orientation estimation, online learning**

I. Introduction

Quaternions are well-established for modelling three-dimensional rotations in computer graphics, navigation, and robotics [1]. In this direction, some of their key advantages include their compact representation, invariance to the so-called gimbal lock, and offer an alternative framework to rotation matrices. Quaternion-valued neural networks have been developed mostly for multidimensional signal processing applications rather than tailored for three-dimensional tracking [2], [3]. For instance, the error formulation in quaternion learning algorithms uses the arithmetic difference, i.e.

$$e(k) = d(k) - y(k) \tag{1}$$

where $d(k)$ and $y(k)$ denote the target and the output of the learning algorithms. This formulation treats quaternions as vectors in $\mathbb{R}^4$, measuring error according to the Euclidean metric. Whilst effective for general quaternion signal processing, this approach is not optimal for orientation estimation which requires quaternions to be constrained to unit norm.

Unit quaternions representing 3D rotations form the manifold $\mathbb{S}^3$—a curved hypersphere embedded in $\mathbb{R}^4$. On this manifold, the geometrically meaningful distance is the geodesic: the arc length of the shortest path along the sphere's surface connecting two orientations. Euclidean distance, which measures straight-line separation through ambient space, does not capture this intrinsic geometry.

The quaternion double-cover presents a further complication: q and $-q$ encode the same physical rotation. To see the rotation identity $q \equiv -q$, consider the quaternion $-q$ and the following quaternion rotation equation.

$$\begin{aligned} p_{\text{n}} &= (-q)p_{\text{o}}(-q)^{-1} & (2)\\ &= (-q)p_{\text{o}}(-q)^{*} \\ &= qp_{\text{o}}q^{*} = qp_{\text{o}}q^{-1} & (3)\end{aligned}$$

where p_{o} and p_{n} denote the old position and the new position resulting from the rotation $-q$ or q on the unit sphere. Eqs. (2) and (3) thus demonstrate the equivalence of $q \equiv -q$ in terms of 3D rotations. This double cover $q \equiv -q$ poses a problem in the error formulation in (1). In other words, if the target is $d(k) = q$ and $y(k) = -q$, the error does *not* go to zero. As such, the spurious error model in (1) disrupts learning when the quaternion sign changes along the trajectory.

Geometric loss functions have been explored for camera pose regression to address representation challenges in orientation estimation [4]. Recent work on dual quaternions has achieved rotational and translational equivariance for 3D rigid motion modelling [5]. These developments motivate a principled geodesic approach for quaternion neural networks that respects the underlying manifold structure.

This work proposes a quaternion neural network with geodesic loss that respects the Riemannian structure of the unit quaternion manifold $\mathbb{S}^3$. The contributions are: (i) a geodesic loss function respecting the manifold geometry; (ii) an error formulation enabling standard quaternion backpropagation; (iii) validation on both synthetic inertial measurement unit (IMU) data and real-world micro aerial vehicle (MAV) flight data.

II. Mathematical Foundations

Prior to the introduction of our proposed method, we revisit briefly the mathematical background required to derive our proposed method.

A. Quaternions and 3D Rotations

A quaternion $q \in \mathbb{H}$ is expressed as:

$$q = q_s + q_x\imath + q_y\jmath + q_z\kappa \tag{4}$$

where $q_s, q_x, q_y, q_z \in \mathbb{R}$ and the imaginary units satisfy $\imath^2 = \jmath^2 = \kappa^2 = \imath\jmath\kappa = -1$. The quaternion conjugate is:

$$q^* = q_s - q_x\imath - q_y\jmath - q_z\kappa \tag{5}$$

This work considers quaternions that represent 3D rotations, which requires two fundamental properties. First, all quaternion variables are assumed to have unit norm, that is, $\|q\| = \sqrt{qq^*} = 1$. This constraint ensures that quaternions lie on the unit hypersphere $\mathbb{S}^3 \subset \mathbb{R}^4$ and allows inversion via conjugation: $q^{-1} = q^*$. Second, unit quaternions admit the Euler form $q = \cos(\theta) + \xi \sin(\theta)$, where $\xi = \Im(q)/\|\Im(q)\|$ is a unit pure quaternion satisfying $\xi^2 = -1$ and $\theta \in [0, \pi]$ represents the rotation half-angle. This representation establishes a correspondence between unit quaternions and points on a complex plane spanned by the real axis and ξ, enabling the use of the quaternion exponential and logarithm for geodesic computation, which is next presented.

B. Geodesic Distance

The natural logarithm for a unit quaternion q behaving as a complex number $\big(\text{i.e. } q = \cos(\theta) + \xi \sin(\theta)\big)$ is given by

$$\ln(q) = \theta\xi \tag{6}$$

where $\theta \in [0, \pi]$ is the rotation half-angle and ξ is the unit pure quaternion corresponding to the rotation axis.

The geodesic distance between unit quaternions p and q can therefore be calculated as [6]:

$$\text{distance}_g(p, q) = \|\ln(p^{-1}q)\| = \|\ln(p^*q)\| \tag{7}$$

where $||\theta\xi|| = \theta$ corresponds to the Euclidean norm of the imaginary (vector) part of the resulting quaternion product p^*q. This distance metric can be shown to measure the angular difference as

$$\text{distance}_g(p, q) = |\theta_q - \theta_p| \tag{8}$$

where θ_x denotes the angle of the corresponding quaternion x. Since q and $-q$ correspond to the same angle, the geodesic distance (8) goes to zero, as desired. Now that the quaternion background has been presented, our proposed quaternion neural network can now be introduced.

III. Quaternion Neural Network with Geodesic Loss

A. Algebraic and notation considerations

Consider that the output of neuron n in layer l at time k is denoted as

$$y_n^{(0)}(k) = x_n(k) \tag{9}$$

$$y_n^{(l)}(k) = \Phi\left(\sum_{m=1}^{N_{l-1}} w_{nm}^{(l)}(k) y_m^{(l-1)}(k) + b_n^{(l)}(k)\right) \tag{10}$$

where $n \in \{1, \ldots, N_l\}$ indexes neurons in layer l, $w_{nm}^{(l)}(k)$ are quaternion weights, $b_n^{(l)}(k)$ are biases, and $\Phi(\cdot)$ is the split activation, that is,

$$\Phi(q) = \tanh(q_s) + \imath \tanh(q_x) + \jmath \tanh(q_y) + \kappa \tanh(q_z) \tag{11}$$

To account for quaternion orientation, the output is normalised to unit norm as

$$\hat{y}(k) = y^{(L)}(k)/\|y^{(L)}(k)\| \tag{12}$$

Algorithm 1 Quaternion Neural Network with Geodesic Loss

Initialisation: $w_{nm}^{(l)}(0) = 0.01 \cdot \mathbf{1}$, $b_n^{(l)}(0) = 0$
For all $k \geq 1$ **Compute forward phase:**
For $l = 1, \ldots, L$ **and** $n = 1, \ldots, N_l$

$$y_n^{(0)}(k) = x_n(k)$$
$$v_n^{(l)}(k) = \sum_{m=1}^{N_{l-1}} w_{nm}^{(l)}(k) y_m^{(l-1)}(k) + b_n^{(l)}(k)$$
$$y_n^{(l)}(k) = \Phi(v_n^{(l)}(k))$$

where

$$\Phi(q) = \tanh(q_s) + \imath \tanh(q_x) + \jmath \tanh(q_y) + \kappa \tanh(q_z)$$
$$\hat{y}(k) = y^{(L)}(k)/\|y^{(L)}(k)\|$$

Compute geodesic error:

$$v_{\text{err}}(k) = \Im(\ln(\hat{y}^*(k)d(k)))$$
$$e_{\text{signal}}(k) = \hat{y}(k)v_{\text{err}}(k)$$

Backpropagate errors:

$$\delta_n^{(l)}(k) = \begin{cases} e_{\text{signal}}(k)\Phi'(v_n^{(l)}(k)) & l = L \\ \left(\displaystyle\sum_{m=1}^{N_{l+1}} \delta_m^{(l+1)}(k) w_{mn}^{(l+1)}(k)\right) \Phi'(v_n^{(l)}(k)) & l \neq L \end{cases}$$

Update weights and bias:

$$w_{nm}^{(l)}(k+1) = w_{nm}^{(l)}(k) + \mu \delta_n^{(l)}(k) \left(y_m^{(l-1)}(k)\right)^*$$
$$b_n^{(l)}(k+1) = b_n^{(l)}(k) + \mu \delta_n^{(l)}(k)$$

where μ is the learning rate and $\Im(q)$ denotes the imaginary part of q.

The geodesic error can be formulated without the norm to enable gradient computation as

$$e(k) = \ln(\hat{y}^*(k)d(k)) \tag{13}$$

Note that the error $e(k)$ is pure imaginary in the context of unit quaternions. The loss function is given by

$$J(k) = \|e(k)\|^2 = \|\ln(\hat{y}^*(k)d(k))\|^2 \tag{14}$$

B. Error Formulation for Backpropagation

The geodesic distance in (7) cannot be used directly as an error signal due to the norm operation. The error in the tangent space at the identity is therefore formulated as follows [6]:

$$v_{\text{err}}(k) = \Im(\ln(\hat{y}^*(k)d(k))) \tag{15}$$

where $\Im(\cdot)$ denotes the vector/imaginary part of the quaternion. Observe that a factor of 2 arising from the half-angle[1]

[1]This half angle stems from the fact that $q = \cos(\vartheta/2) + \xi \sin(\vartheta/2)$, leading to the logarithm in (6) yielding the half angle $\vartheta/2\xi$.

representation of quaternion rotations is omitted in (15). This is because, it can be compensated (or absorbed) in the learning rate. To project the error from the reference point to the current position onto the sphere, we have:

$$e_{\text{signal}}(k) = \hat{y}(k)v_{\text{err}}(k) \tag{16}$$

To justify (16), note that $v_{\text{err}}(k)$ is pure imaginary and therefore lies in the tangent space at the identity, $T_{\mathbf{1}}\mathbb{S}^3$, whereas backpropagation requires the error at the current output, i.e., in $T_{\hat{y}(k)}\mathbb{S}^3$. Since $\mathbb{S}^3$ is a Lie group under quaternion multiplication, left translation by $\hat{y}(k)$ maps between these tangent spaces, and under the bi-invariant metric this coincides with parallel transport along the geodesic from $\mathbf{1}$ to $\hat{y}(k)$ [7]. The product $\hat{y}(k)v_{\text{err}}(k)$ in (16) is thus the transported error, enabling standard quaternion backpropagation whilst preserving the geodesic information.

The local gradient for neuron n in layer l can be computed as

$$\delta_n^{(l)}(k) = \begin{cases} e_{\text{signal}}(k)\Phi'(v_n^{(l)}(k)) & l = L \\ \left(\sum_m \delta_m^{(l+1)}(k)w_{mn}^{(l+1)}(k)\right)\Phi'(v_n^{(l)}(k)) & l < L \end{cases} \tag{17}$$

where $v_n^{(l)}(k)$ is the 'net' value and

$$\begin{aligned}\Phi'(v) =& (1-\tanh^2(v_s)) + \imath(1-\tanh^2(v_x)) \\ &+ \jmath(1-\tanh^2(v_y)) + \kappa(1-\tanh^2(v_z))\end{aligned} \tag{18}$$

For the output neuron at the final layer, the error signal $e_{\text{signal}}(k)$ from (16) was considered as it is. Weights are updated in an online fashion as

$$w_{nm}^{(l)}(k+1) = w_{nm}^{(l)}(k) + \mu\delta_n^{(l)}(k)(y_m^{(l-1)}(k))^* \tag{19}$$

$$b_n^{(l)}(k+1) = b_n^{(l)}(k) + \mu\delta_n^{(l)}(k) \tag{20}$$

where μ is the learning rate. For ease of implementation, Algorithm 1 summarises our proposed method.

IV. Experiments

To assess the efficacy of our quaternion neural network, we considered the real-world problem of orientation estimation from the measurements of inertial measurement units (IMU), i.e. acceleration (a) and angular velocity (ω). Moreover, this section compared our proposed method against the traditional feedforward quaternion neural network (referred as 'Euclidean baseline') in two sets of experiments. The first one involved controlled data generated from spherical linear interpolation between two quaternion rotations, whereas the second set of experiment considered real-world data of an aerial vehicle. Both networks are trained online, with each sample predicted before being used to update the weights. The reported performance therefore reflects the one-step-ahead tracking error across the entire sequence, which is the setting of interest for real-time orientation estimation on streaming IMU data.

A. Datasets

1) Synthetic IMU Data: Synthetic data comprising 6,000 samples of continuous 3D rotations was generated. Reference orientations were created using spherical linear interpolation (SLERP) between random keyframe quaternions [8], providing smooth rotation paths. Accelerometer and gyroscope readings were synthesised from orientations with additive Gaussian noise ($\sigma_a = 0.1$, $\sigma_\omega = 0.01$).

2) EuRoC MAV Dataset: Real-world validation used the EuRoC Micro Aerial Vehicle (MAV) dataset [9], a widely-used benchmark for visual-inertial odometry. The dataset was recorded onboard a MAV flying in indoor environments, providing stereo camera images and synchronised inertial measurements. Ground truth poses were obtained using a Vicon motion capture system at millimetre precision. The IMU measurements contain 3-axis gyroscope and accelerometer data sampled at 200 Hz. We used the MH_01_easy sequence, which contains 3,682 samples of MAV flight including hover, forward flight, and turning manoeuvres.

B. Network Configuration

Both the geodesic method and Euclidean baseline used the network architecture [2,3,3,1]: two quaternion inputs, two hidden layers with three neurons, and one quaternion output. The two inputs were fed with the accelerometer and gyroscope readings encoded as pure quaternions, i.e., $x_1(k) = a_x\imath + a_y\jmath + a_z\kappa$ and $x_2(k) = \omega_x\imath + \omega_y\jmath + \omega_z\kappa$. A learning rate of $\mu = 0.05$ was selected via grid search over $\mu \in [0.01, 0.1]$. Target trajectories were preprocessed to ensure consecutive quaternions have positive inner product, addressing the quaternion double-cover and preventing discontinuous jumps. For performance evaluation, the angular error metric considered is given by

$$\epsilon_{\text{angle}}(k) = 2\arccos(|\langle\hat{y}(k), d(k)\rangle|) \tag{21}$$

where $\langle\cdot,\cdot\rangle$ denotes the quaternion inner product.

C. Results

Fig. 1 shows the results on synthetic IMU data. On its left, the error plot demonstrates that whilst the Euclidean method exhibited large spikes reaching 14° during rapid orientation changes, the geodesic method maintained stable, low error throughout with maximum error of only 0.32°. The right sphere visualisation confirms that the geodesic (blue) method followed the ground truth accurately whilst the Euclidean (green) method deviated during transitions.

Fig. 2 presents the results on the EuRoC MAV dataset. On the right plot, the geodesic (blue) method successfully tracked orientation during indoor flight including hover, forward motion, and turning manoeuvres. On the left plot, it is clear that the Euclidean (green) method exhibited substantial errors, particularly during rapid rotations. The consistent performance of the geodesic method on real-world sensor data demonstrates its robustness against noise and flight dynamics.

In addition to the qualitative results in Figs. 1-2, Table I compares the tracking performances from a quantitative perspective. The geodesic method achieved consistently superior

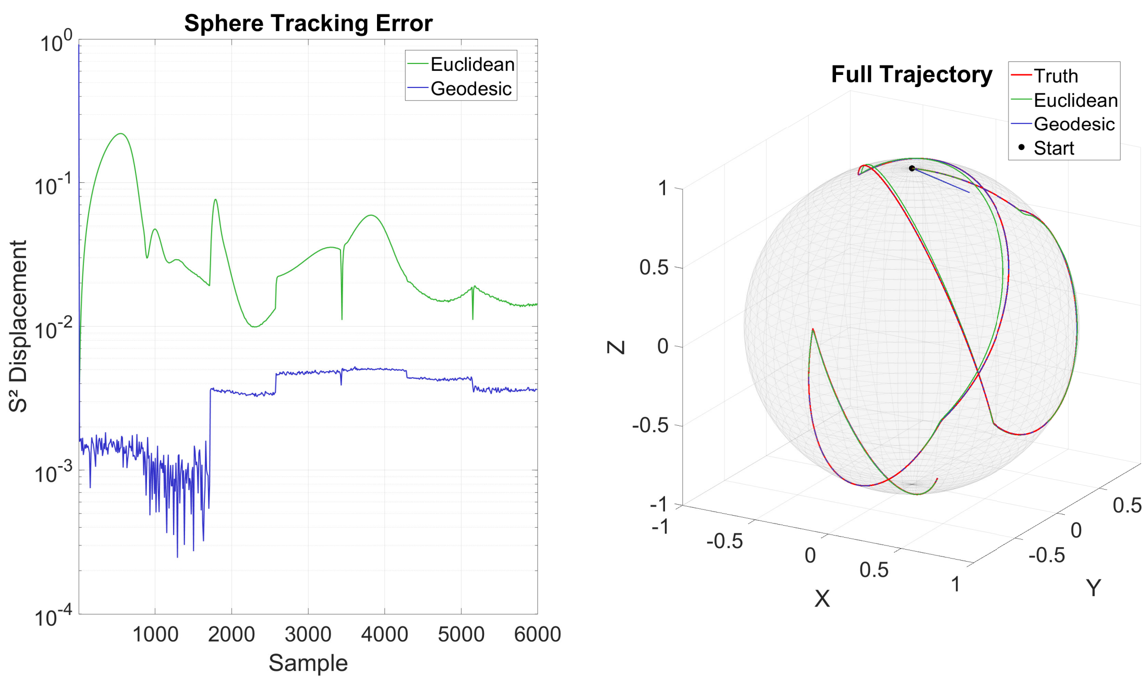

Fig. 1: Synthetic IMU data results. (a) Angular error (log scale). Geodesic (blue) maintains stable low error; Euclidean (green) shows large spikes. (b) Trajectory on $\mathbb{S}^2$. Geodesic tracks ground truth (red) accurately.

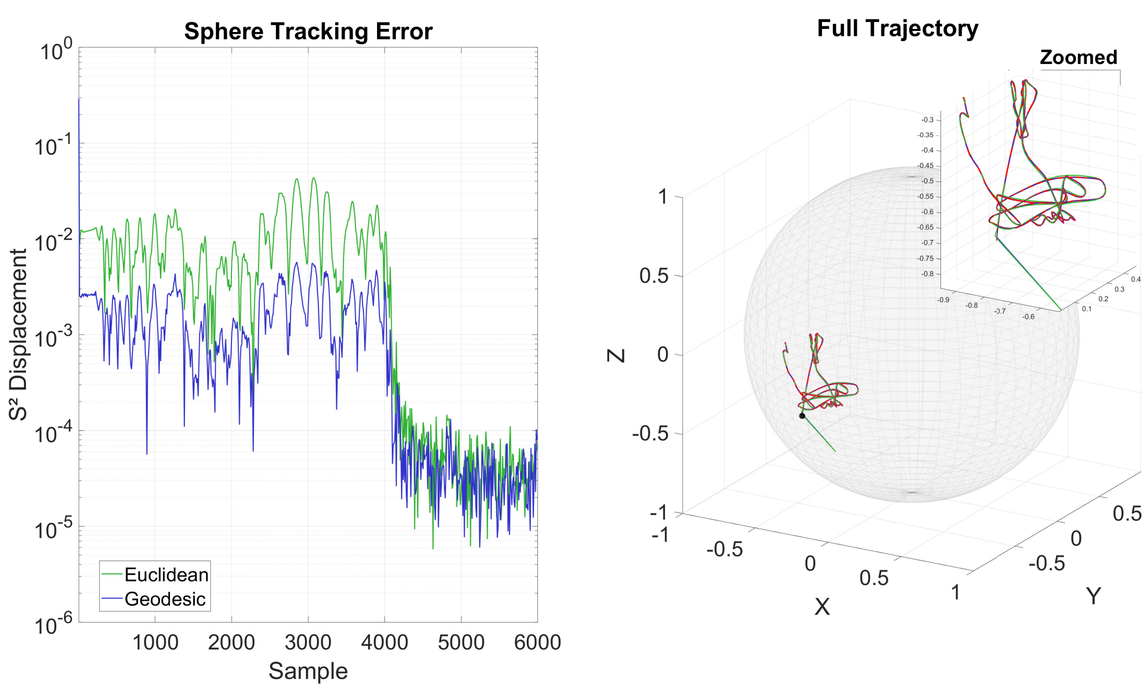

Fig. 2: EuRoC MAV dataset results. (a) Angular error during real flight manoeuvres. (b) Trajectory on $\mathbb{S}^2$ during indoor MAV flight, validating practical applicability on real sensor data.

TABLE I: Orientation Tracking Performance

Dataset	Method	Mean (°)	Max (°)	Std (°)
Synthetic	Euclidean	2.25	14.09	2.41
	Geodesic	**0.22**	**0.32**	**0.05**
EuRoC [9]	Euclidean	0.45	2.52	0.53
	Geodesic	**0.08**	**0.37**	**0.08**

performance in both synthetic and real-world data, demonstrating significant improvement over the Euclidean baseline.

V. Discussion

The consistent performance improvement across both synthetic and real EuRoC data demonstrates the importance of incorporating the manifold geometry. The Euclidean error treats quaternions as points in $\mathbb{R}^4$, ignoring that they lie on the 3-sphere with non-trivial curvature. In orientation tracking, geodesic distance is physically meaningful as it measures angular differences directly.

The error formulation in (15)–(16) bridges geodesic optimisation and standard quaternion backpropagation without requiring Riemannian gradient computations. The online formulation updates weights at each time step without storing past samples, making the approach suitable for edge computing devices.

VI. Conclusion

In this work, we introduced a quaternion-valued neural network equipped with a geodesic loss function that respects the intrinsic curved geometry of the hypersphere $\mathbb{S}^3$. By formulating the loss directly on the rotation manifold and explicitly addressing the quaternion sign ambiguity $q \equiv -q$, the proposed approach provides a theoretically consistent framework for learning 3D orientations. Our experimental results on both synthetic and real-world datasets show that the performance advantage of our geometrically grounded formulation over conventional Euclidean-based methods for orientation estimation. These findings highlight the importance of incorporating manifold-aware error metrics in quaternion learning and establish geodesic-based loss functions as a powerful direction for future research in rotation estimation and geometric deep learning.

References

[1] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press, 2020.

[2] T. Parcollet, M. Ravanelli, M. Morchid, G. Linarès, C. Trabelsi, R. De Mori, and Y. Bengio, “Quaternion recurrent neural networks,” in *Proc. ICLR*, 2019.

[3] C. Cheong Took, S. P. Talebi, R. M. Fernandez Alcala, and D. P. Mandic, “Augmented statistics of quaternion random variables,” *IEEE Signal Process. Mag.*, vol. 41, no. 3, pp. 72–87, 2024.

[4] A. Kendall and R. Cipolla, “Geometric loss functions for camera pose regression,” in *Proc. CVPR*, 2017, pp. 5974–5983.

[5] G. Vieira, E. Grassucci, M. E. Valle, and D. Comminiello, “Dual quaternion rotational and translational equivariance in 3D rigid motion modelling,” in *Proc. IEEE 33rd Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Rome, Italy, 2023, honorable Mention: Top 10% Outstanding Paper.

[6] F. C. Park and B. Ravani, “Smooth invariant interpolation of rotations,” *ACM Trans. Graph.*, vol. 16, no. 3, pp. 277–295, 1997.

[7] J. M. Lee, *Introduction to Riemannian Manifolds*, 2nd ed. New York, NY, USA: Springer, 2018.

[8] K. Shoemake, “Animating rotation with quaternion curves,” *ACM SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 245–254, 1985.

[9] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The EuRoC micro aerial vehicle datasets,” *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.