Dual Quaternion Particle Filtering for Pose Estimation

Aksel Sveier, *Member, IEEE*, and Olav Egeland, *Senior Member, IEEE*

***Abstract*— This article presents a particle filter for pose estimation using unit dual quaternion kinematics. The eight-parameter unit dual quaternion is used for global representation of the pose, whereas the six parameters of the dual modified Rodrigues parameters (MRPs) are used for local pose representation in the state-space model. The dual MRPs enable estimates of the mean and the covariance to be calculated from the particles without violating the algebraic constraint of the unit dual quaternion. For verification of the filter and comparison with state of the art, we consider pose measurements available in the form of unit dual quaternions. Angular velocity and specific force measurements from a body-mounted inertial measurement unit are also considered in the filtering. We show through simulations that the suggested particle filter has comparable accuracy with a previously proposed unscented Kalman filter based on unit dual quaternions. We also consider a practical application where the pose of an arbitrary rigid object is estimated using a sequence of point clouds from a 3-D camera. A model point cloud of the object is displaced with the unit dual quaternion associated with each particle, and a fitting score is calculated between the displaced model point cloud and the measured point cloud from the 3-D camera. The likelihoods of the fitting scores are calculated from an exponential distribution and are used to update the weights of the particles. The system was verified in an experiment where the motion of a swinging payload hanging from a crane was estimated using a 3-D camera.**

***Index Terms*— Dual quaternions, kalman filters, nonlinear filters, particle filtters, pose estimation.**

I. Introduction

A MULTIPLICATIVE extended Kalman filter (EKF) for attitude estimation based on unit quaternions was presented in [1]. The unit quaternion is a four-parameter global parameterization of attitude, where the parameters appear linearly in the kinematic equation of motion. The multiplicative measurement update of the EKF ensures the unit constraint of the quaternion. A survey of attitude parameterizations was presented in [2], while nonlinear filtering methods for attitude estimation were surveyed in [3]. It was concluded that the unit quaternion was the most widely used parameterization in attitude estimation, due to the aforementioned properties.

A unit dual quaternion is an eight-parameter global parameterization of pose. The kinematic equation of motion describes the simultaneous motion of translation and rotation and has a similar structure as the kinematic equation of the unit quaternion. EKFs with unit dual quaternions in the state were developed in [4] and [5], where both methods used additive measurement updates, which meant that the algebraic constraint of the unit dual quaternion would not hold without a normalization of the updates. The multiplicative approach for attitude estimation with unit quaternions in [1], which ensured that the algebraic constraint would hold, was extended to a multiplicative EKF for pose estimation with unit dual quaternions in [6].

An unscented Kalman filter (UKF) for attitude estimation with unit quaternion kinematics was presented in [7]. Modified Rodrigues parameters (MRPs) were used for local attitude representation in the state, whereas unit quaternions were used for global representation. With this approach, the mean and covariance of the state could be obtained from the sigma points of the UKF without violating the algebraic constraint of the unit quaternion. These results were extended to UKF for pose estimation with unit dual quaternions in [8]. The unit dual quaternion was transformed into the six parameter twistor representation [9] in the state, where the twistor corresponds to a dual MRP.

Particle filters require methods for finding estimates such as the weighted mean and covariance from a set of weighted state particles. This is analogous to calculating the mean and covariance from sigma points in the UKF. The results in [7] were used to develop a particle filter for attitude estimation with unit quaternions in [10], where the local MRPs were used in the particle states. Unlike EKFs and UKFs, the particle filter does not assume that the state is distributed with a Gaussian probability density function (pdf). Instead, a large number of weighted samples, drawn from the state space, approximate the continuous pdf of the state. This approach assumes no functional form of the pdf and can solve nonlinear and non-Gaussian filtering problems [11].

Particle filters are especially suited in combination with camera sensor systems, as retrieving pose measurements from visual data results in nonlinear and non-Gaussian operations. In [12], a likelihood function with an exponential distribution was used in a particle filter to estimate the pose of an object in consecutive images from a 2-D camera. A known wireframe of the object was projected into the image using the particle pose. In [13], a particle filter was used to estimate the pose of objects in a 3-D camera field of view. The 3-D images were represented as point clouds, and point cloud models of the objects were evaluated with an exponential likelihood function. In general, clutter and occlusions in visual data may

Manuscript received December 14, 2018; revised May 5, 2020; accepted September 4, 2020. Date of publication October 14, 2020; date of current version August 5, 2021. Manuscript received in final form September 23, 2020. This work was supported by the Norwegian Research Council, SFI Offshore Mechatronics, under Project 237896. Recommended by Associate Editor J.-S. Li. *(Corresponding author: Aksel Sveier.)*

The authors are with the Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway (e-mail: akselsveier@gmail.com; olav.egeland@ntnu.no).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2020.3026926

also cause ambiguities in the measurements, causing the state to be distributed with a multimodal pdf. A classical example is found in [14], where curves in 2-D images were tracked in a cluttered environment using a particle filter.

High-rate accelerometers and gyroscopes can be used as complementary measurements to low-rate visual measurements. In [6] and [8], these sensor measurements were used in the prediction step of the filter and were modeled as control inputs. The sensors provide angular velocity and specific force measurements, which are subject to drift that will accumulate in the integration of the signals. It is therefore imperative to correct the state estimate in the update step of the filter using a drift-free measurement system, as discussed in detail in [15].

In this article, we extend the particle filter for attitude estimation using unit quaternions in [10] to particle filtering for pose estimation with unit dual quaternions. Similar to the UKF solution in [8], we use dual MRPs for local error representation in the state vector, whereas unit dual quaternions are used for the global representation. Moreover, we use the Cayley transform [16] to perform the mapping between the unit dual quaternion representation and the dual MRP representation. The Cayley transform is also identified as an approximation of the exponential function for dual vectors and is used in the discrete-time propagation of the system kinematics. The prediction step of the particle filter is modeled with angular and linear velocity measurements that are corrupted by bias and noise. We show how the same model can be used when measurements of angular and linear velocity are unavailable. We also show how to include specific force measurements from an accelerometer with bias and noise by replacing the linear velocity measurements. In the measurement update, we first consider the case where pose measurements are available in the form of unit dual quaternions and use the measurement relation suggested in [17]. This is used for verification of the filter through simulations, including a comparison with the dual quaternion UKF presented in [8]. To show how the proposed particle filter can be used in combination with a vision system, we then consider point cloud measurements from a 3-D camera. The likelihood of each particle is calculated from an exponential distribution using a fitting score. This solution was verified in an experiment where the motion of a swinging payload hanging from a crane was estimated using a 3-D camera. A separate Aruco marker system was used for ground-truth reference in the experiment.

This article is organized as follows. Section II presents the preliminary results on quaternions, dual number theory, and kinematics, and the general particle filter equations are introduced. Section III presents the dual quaternion particle filter with the prediction and measurement update equations and a discussion on implementation considerations. Section IV shows the simulation results of the particle filter, and Section V presents the swinging payload experiment where the particle filter was used for tracking the motion. Finally, this article is concluded in Section VI.

II. Preliminaries

In this section, we present the required background for quaternions and dual quaternions. This includes a discussion on exponential mappings and logarithms, a presentation of the Cayley transform, which can be used to compute the MRPs, and an approximation of the exponential mapping. In addition, the particle filter equations are presented.

A. Quaternions

A quaternion can be represented as a sum $\mathbf{q} = \alpha + \boldsymbol{\beta}$ of a scalar $\alpha \in \mathbb{R}$ and a vector $\boldsymbol{\beta} \in \mathbb{R}^3$ [18], [19]. The conjugate of the quaternion is given by $\mathbf{q}^* = \alpha - \boldsymbol{\beta}$, and multiplication with a scalar λ gives $\lambda\mathbf{q} = \lambda\alpha + \lambda\boldsymbol{\beta}$. Addition and subtraction of two quaternions $\mathbf{q}_1 = \alpha_1 + \boldsymbol{\beta}_1$ and $\mathbf{q}_2 = \alpha_2 + \boldsymbol{\beta}_2$ are done componentwise and give $\mathbf{q}_1 \pm \mathbf{q}_2 = (\alpha_1 \pm \alpha_2) + (\boldsymbol{\beta}_1 \pm \boldsymbol{\beta}_2)$. The quaternion product is denoted with $\otimes$ and is defined by $\mathbf{q}_1 \otimes \mathbf{q}_2 = (\alpha_1\alpha_2 - \boldsymbol{\beta}_1 \cdot \boldsymbol{\beta}_2) + (\alpha_1\boldsymbol{\beta}_2 + \alpha_2\boldsymbol{\beta}_1 + \boldsymbol{\beta}_1 \times \boldsymbol{\beta}_2)$, where $\cdot$ denotes the scalar product and $\times$ denotes the vector cross product. A vector $\mathbf{u}$ can be treated as a quaternion with zero scalar part. It follows that $\mathbf{u}^* = -\mathbf{u}$ and that $\mathbf{q} \otimes \mathbf{u} = -\boldsymbol{\beta} \cdot \mathbf{u} + (\alpha\mathbf{u} + \boldsymbol{\beta} \times \mathbf{u})$. The magnitude of a quaternion is $\|\mathbf{q}\|^2 = \mathbf{q} \otimes \mathbf{q}^* = \alpha^2 + \boldsymbol{\beta} \cdot \boldsymbol{\beta}$ and the inverse is given by $\mathbf{q}^{-1} = \mathbf{q}^*/\|\mathbf{q}\|^2$. The quaternion can also be represented as a column vector $[\mathbf{q}] = [\alpha, \ \boldsymbol{\beta}^\top]^\top$. Then, the magnitude is given by $\|\mathbf{q}\|^2 = [\mathbf{q}]^\top[\mathbf{q}]$.

A unit quaternion $\mathbf{q} = \eta + \boldsymbol{\sigma}$ is a quaternion with unit magnitude $\|\mathbf{q}\|^2 = \eta^2 + \boldsymbol{\sigma} \cdot \boldsymbol{\sigma} = 1$. A rotation θ about the unit axis $\mathbf{k}$ through the origin can be expressed as a unit quaternion in the form $\mathbf{q} = \cos\theta/2 + \sin\theta/2\mathbf{k}$. The corresponding rotation matrix is given by $\mathbf{R}(\mathbf{q}) = \mathbf{I} + 2\eta\boldsymbol{\sigma}^\times + 2\boldsymbol{\sigma}^\times\boldsymbol{\sigma}^\times$, where $(\cdot)^\times$ denotes the skew symmetric form of a column vector and $\mathbf{I}$ is the identity matrix. The composite rotation $\mathbf{R} = \mathbf{R}_1\mathbf{R}_2$ is a rotation $\mathbf{R}_1$ followed by a rotation $\mathbf{R}_2$. Then, if $\mathbf{R}_1$ corresponds to the unit quaternion $\mathbf{q}_1$ and $\mathbf{R}_2$ corresponds to the unit quaternion $\mathbf{q}_2$, the composite rotation $\mathbf{R}$ will correspond to the unit quaternion $\mathbf{q} = \mathbf{q}_1 \otimes \mathbf{q}_2$. It is noted that the quaternion product of two unit quaternions is a unit quaternion, while the sum of two unit quaternions will be a quaternion that in general will not be a unit quaternion.

The unit quaternion can be expressed in terms of the exponential function

$$\mathbf{q} = \exp\left(\frac{\theta\mathbf{k}}{2}\right) = 1 + \frac{\theta\mathbf{k}}{2} + \frac{1}{2!}\left(\frac{\theta\mathbf{k}}{2}\right)^2 + \frac{1}{3!}\left(\frac{\theta\mathbf{k}}{2}\right)^3 + \cdots$$

where $(\cdot)^n$ is the quaternion product of order n. The vector $\mathbf{u} = \theta\mathbf{k}/2$ is therefore called the logarithm of $\mathbf{q}$. If the logarithm $\mathbf{u}$ is given, then $\|\mathbf{u}\| = \pm\theta/2$ and $\mathbf{u}/\|\mathbf{u}\| = \pm\mathbf{k}$. Although the expression for $\mathbf{k}$ is undefined for $\mathbf{u} = \mathbf{0}$, the exponential can be calculated for all $\mathbf{u}$ from

$$\exp(\mathbf{u}) = \cos(\|\mathbf{u}\|) + \operatorname{sinc}(\|\mathbf{u}\|)\mathbf{u}.$$

A similar expression was used in [20] to calculate the rotation matrix from the logarithm. This expression is numerically well-conditioned around $\mathbf{u} = \mathbf{0}$, which is seen from the Taylor series expansion $\operatorname{sinc}(x) = \sin x/x = 1 - x^2/3! + x^4/5! + \cdots$

B. Dual Numbers

A dual number $\tilde{a}$ is given by $\tilde{a} = a + \varepsilon a'$, where a and a' are real numbers and ε is the dual unit defined by $\varepsilon \neq 0$ and $\varepsilon^2 = 0$ [21], [22]. Multiplication with a scalar λ is given by $\lambda\tilde{a} = \lambda a + \varepsilon\lambda a'$. Addition and subtraction of two dual

numbers $\tilde{a}_1 = a_1 + \varepsilon a_1'$ and $\tilde{a}_2 = a_2 + \varepsilon a_2'$ are componentwise and given by $\tilde{a}_1 \pm \tilde{a}_2 = a_1 \pm a_2 + \varepsilon(a_1' \pm a_2')$. Multiplication is done by regarding $\tilde{a}_1$ and $\tilde{a}_2$ as polynomials in ε and using $\varepsilon^2 = 0$. This gives $\tilde{a}_1\tilde{a}_2 = a_1a_2 + \varepsilon(a_1a_2' + a_1'a_2)$. The inverse is $\tilde{a}^{-1} = 1/\tilde{a} = (1/a) - \varepsilon a'/a^2$. The real-valued function $f(a)$ of a is extended to a dual function $f(\tilde{a})$ by a Taylor series expansion about a, which in view of $\varepsilon^2 = 0$ gives

$$f(\tilde{a}) = f(a) + \varepsilon a' \frac{\mathrm{d}f(a)}{\mathrm{d}a}. \tag{1}$$

Let θ be an angle and let d be a scalar distance. Then, the dual number $\tilde{\theta} = \theta + \varepsilon d$ is called a dual angle. It follows from (1) that the sine and cosine of the dual angle are:

$$\sin\tilde{\theta} = \sin\theta + \varepsilon d\cos\theta$$
$$\cos\tilde{\theta} = \cos\theta - \varepsilon d\sin\theta.$$

It is straightforward to verify that $\sin^2\tilde{\theta} + \cos^2\tilde{\theta} = 1$ and that the usual trigonometric identities $\sin\tilde{\theta} = 2\sin(\tilde{\theta}/2)\cos(\tilde{\theta}/2)$ and $\cos\tilde{\theta} = \cos^2(\tilde{\theta}/2) - \sin^2(\tilde{\theta}/2)$ are satisfied. Moreover, the Taylor series expansions about $\tilde{\theta} = 0$ are

$$\sin\tilde{\theta} = \tilde{\theta} - \frac{\tilde{\theta}^3}{3!} + \frac{\tilde{\theta}^5}{5!} + \cdots$$
$$\cos\tilde{\theta} = 1 - \frac{\tilde{\theta}^2}{2!} + \frac{\tilde{\theta}^4}{4!} + \cdots$$

It is noted that $\tan\tilde{\theta} = \sin\tilde{\theta}/\cos\tilde{\theta} = \tan\theta + \varepsilon(d/\cos^2\theta)$.

A dual vector $\tilde{\mathbf{u}}$ is given by $\tilde{\mathbf{u}} = \mathbf{u} + \varepsilon\mathbf{u}'$, where $\mathbf{u}$ and $\mathbf{u}'$ are vectors. The scalar product of two dual vectors $\tilde{\mathbf{u}}_1 = \mathbf{u}_1 + \varepsilon\mathbf{u}_1'$ and $\tilde{\mathbf{u}}_2 = \mathbf{u}_2 + \varepsilon\mathbf{u}_2'$ is $\tilde{\mathbf{u}}_1 \cdot \tilde{\mathbf{u}}_2 = \mathbf{u}_1 \cdot \mathbf{u}_2 + \varepsilon(\mathbf{u}_1 \cdot \mathbf{u}_2' + \mathbf{u}_1' \cdot \mathbf{u}_2)$, while the cross product is $\tilde{\mathbf{u}}_1 \times \tilde{\mathbf{u}}_2 = \mathbf{u}_1 \times \mathbf{u}_2 + \varepsilon(\mathbf{u}_1 \times \mathbf{u}_2' + \mathbf{u}_1' \times \mathbf{u}_2)$.

A line can be written as a dual vector $\tilde{\mathbf{k}} = \mathbf{k} + \varepsilon\mathbf{k}'$, where $\mathbf{k} \in \mathbb{R}^3$ is a unit vector along the line, $\mathbf{k}' = \mathbf{r} \times \mathbf{k}$ is the moment of the line with respect to a reference point, and $\mathbf{r} \in \mathbb{R}^3$ is the vector from the reference point to a point on the line. It is noted that $\mathbf{k} \cdot \mathbf{k}' = 0$. The elements $\mathbf{k}$ and $\mathbf{k}'$ are the Plücker coordinates [22] of the line.

C. Dual Quaternions

A dual quaternion is given by $\tilde{\mathbf{q}} = \mathbf{q} + \varepsilon\mathbf{q}'$, where the real part $\mathbf{q} = \alpha + \boldsymbol{\beta}$ and the dual part $\mathbf{q}' = \alpha' + \boldsymbol{\beta}'$ are quaternions [18], [23]. The dual quaternion can also be represented by $\tilde{\mathbf{q}} = \tilde{\alpha} + \tilde{\boldsymbol{\beta}}$, where $\tilde{\alpha} = \alpha + \varepsilon\alpha'$ is a dual scalar and $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta} + \varepsilon\boldsymbol{\beta}'$ is a dual vector. It follows that addition and subtraction of dual quaternions are componentwise. Conjugation is given by $\tilde{\mathbf{q}}^* = \mathbf{q}^* + \varepsilon\mathbf{q}'^* = \tilde{\alpha} - \tilde{\boldsymbol{\beta}}$. The quaternion product of two dual quaternions is given by $\tilde{\mathbf{q}}_1 \otimes \tilde{\mathbf{q}}_2 = \mathbf{q}_1 \otimes \mathbf{q}_2 + \varepsilon(\mathbf{q}_1 \otimes \mathbf{q}_2' + \mathbf{q}_1' \otimes \mathbf{q}_2)$. A dual vector $\tilde{\mathbf{u}} = \mathbf{u} + \varepsilon\mathbf{u}'$ can be treated as a dual quaternion with zero scalar parts. The dual magnitude of a dual quaternion is

$$\|\tilde{\mathbf{q}}\|^2 = \tilde{\mathbf{q}} \otimes \tilde{\mathbf{q}}^* = \|\mathbf{q}\|^2 + 2\varepsilon(\alpha\alpha' + \boldsymbol{\beta} \cdot \boldsymbol{\beta}'). \tag{2}$$

Note that the dual magnitude is a dual number. It follows that the inverse is $\tilde{\mathbf{q}}^{-1} = \tilde{\mathbf{q}}^*/\|\tilde{\mathbf{q}}\|^2$, which can be expressed as

$$(\mathbf{q} + \varepsilon\mathbf{q}')^{-1} = \mathbf{q}^{-1} - \varepsilon\mathbf{q}^{-1} \otimes \mathbf{q}' \otimes \mathbf{q}^{-1} \tag{3}$$

and can be verified by direct calculation. A dual quaternion can also be represented by the column vector $[\tilde{\mathbf{q}}] = [[\mathbf{q}]^\top,\ [\mathbf{q}']^\top]^\top$.

A dual quaternion $\tilde{\mathbf{q}} = \mathbf{q} + \varepsilon\mathbf{q}'$ is a unit dual quaternion if it has unit dual magnitude $\|\tilde{\mathbf{q}}\|^2 = 1 + \varepsilon 0$. From (2), it follows that $\tilde{\mathbf{q}}$ is a unit dual quaternion if and only if $\|\mathbf{q}\|^2 = 1$ and $\alpha\alpha' + \boldsymbol{\beta} \cdot \boldsymbol{\beta}' = 0$. A unit dual quaternion can be used to represent the displacement

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

of a rigid body [22]. The displacement $\mathbf{T}$ can be seen as a composite displacement given by a translation $\mathbf{t}$ followed by a rotation $\mathbf{R}$. Let $\tilde{\mathbf{q}}_R = \mathbf{q}$ be a dual quaternion corresponding to a pure rotation, and let $\tilde{\mathbf{q}}_t = 1 + \varepsilon(1/2)\mathbf{t}$ be the unit dual quaternion representing the translation $\mathbf{t}$. Then, the dual quaternion corresponding to $\mathbf{T}$ will be given by $\tilde{\mathbf{q}} = \tilde{\mathbf{q}}_t \otimes \tilde{\mathbf{q}}_R = \mathbf{q} + \varepsilon(1/2)\mathbf{t} \otimes \mathbf{q}$. It follows from $\mathbf{q}' = (1/2)\mathbf{t} \otimes \mathbf{q}$ that $\mathbf{t} = 2\mathbf{q}' \otimes \mathbf{q}^*$, which gives

$$\mathbf{T}(\tilde{\mathbf{q}}) = \begin{bmatrix} \mathbf{R}(\mathbf{q}) & 2\mathbf{q}' \otimes \mathbf{q}^* \\ \mathbf{0}^\top & 1 \end{bmatrix}.$$

An exponential function of the unit dual quaternion can be defined if the translation and rotation commute. According to Chasles's theorem, a displacement $\mathbf{T}$ can be given as a rotation by an angle θ about a line $\tilde{\mathbf{k}} = \mathbf{k} + \varepsilon\mathbf{k}'$ and a translation d along the same line [24], which means that the translation and rotation commute. The dual angle of this displacement is defined as $\tilde{\theta} = \theta + \varepsilon d$. The unit dual quaternion describing the displacement is

$$\tilde{\mathbf{q}} = \cos\frac{\tilde{\theta}}{2} + \sin\frac{\tilde{\theta}}{2}\tilde{\mathbf{k}} \tag{4}$$

which gives

$$\tilde{\mathbf{q}} = \cos\frac{\theta}{2} + \mathbf{k}\sin\frac{\theta}{2} + \varepsilon\left(-\frac{d}{2}\sin\frac{\theta}{2} + \mathbf{k}\frac{d}{2}\cos\frac{\theta}{2} + \mathbf{k}'\sin\frac{\theta}{2}\right). \tag{5}$$

It is seen that the real part $\mathbf{q}$ is the unit quaternion of the rotation, whereas the dual part $\mathbf{q}'$ is a quaternion, including the parameters of both rotation and translation. Moreover, for a pure rotation where $d = 0$, the unit dual quaternion is $\tilde{\mathbf{q}}_R = \cos\theta/2 + \tilde{\mathbf{k}}\sin\theta/2$, whereas for a pure translation where $\theta = 0$, the unit dual quaternion is $\tilde{\mathbf{q}}_t = 1 + \varepsilon\mathbf{k}(d/2)$.

The unit dual quaternion can be expressed by the exponential function

$$\tilde{\mathbf{q}} = \exp\left(\frac{\tilde{\theta}\tilde{\mathbf{k}}}{2}\right) = 1 + \frac{\tilde{\theta}\tilde{\mathbf{k}}}{2} + \frac{1}{2!}\left(\frac{\tilde{\theta}\tilde{\mathbf{k}}}{2}\right)^2 + \frac{1}{3!}\left(\frac{\tilde{\theta}\tilde{\mathbf{k}}}{2}\right)^3 + \cdots$$

The inverse function

$$\log(\tilde{\mathbf{q}}) = \frac{\tilde{\theta}\tilde{\mathbf{k}}}{2} = \frac{\theta\mathbf{k}}{2} + \varepsilon\left(\frac{d\mathbf{k}}{2} + \frac{\theta\mathbf{k}'}{2}\right)$$

is the dual vector logarithm of $\tilde{\mathbf{q}}$. If the dual vector $\tilde{\mathbf{u}} = \mathbf{u} + \varepsilon\mathbf{u}'$ is given, the exponential function can be calculated from the equivalent expression (5) using $\theta/2 = \|\mathbf{u}\|$, $\mathbf{k} = \mathbf{u}/\|\mathbf{u}\|$, $d/2 = \mathbf{u}^\top\mathbf{u}'/\|\mathbf{u}\|$, and $\mathbf{k}' = -\mathbf{u}^\times\mathbf{u}^\times\mathbf{u}'/\|\mathbf{u}\|^3$. The expressions for $d/2$, $\mathbf{k}$, and $\mathbf{k}'$ are not defined for $\mathbf{u} = \mathbf{0}$, and still the exponential can be calculated from [25]

$$\exp(\tilde{\mathbf{u}}) = \cos(\|\mathbf{u}\|) + \mathrm{sinc}(\|\mathbf{u}\|)\mathbf{u} + \varepsilon\left(-\mathrm{sinc}(\|\mathbf{u}\|)\mathbf{u}^\top + \cos\|\mathbf{u}\|\mathbf{I} + \frac{\|\mathbf{u}\|\cos\|\mathbf{u}\| - \sin\|\mathbf{u}\|}{\|\mathbf{u}\|^3}\mathbf{u}^\times\mathbf{u}^\times\right)\mathbf{u}'. \tag{6}$$

It follows from the Taylor series expansion $(x\cos x - \sin x)/x^3 = -1/3 + x^2/30 + \cdots$ at $x = 0$ that this expression is numerically well conditioned for all $\tilde{\mathbf{u}}$. It is noted that a similar technique was used in [20] to calculate the homogeneous transformation matrix from a logarithm.

D. Cayley Transform

The Cayley transform of the vector $\mathbf{u} \in \mathbb{R}^3$ is given by

$$\operatorname{cay}(\mathbf{u}) \triangleq (1+\mathbf{u}) \otimes (1-\mathbf{u})^{-1}.$$

This expression can be calculated to be

$$\operatorname{cay}(\mathbf{u}) = \frac{(1+\mathbf{u}) \otimes (1+\mathbf{u})}{1+\|\mathbf{u}\|^2} = \frac{1-\|\mathbf{u}\|^2}{1+\|\mathbf{u}\|^2} + \frac{2}{1+\|\mathbf{u}\|^2}\mathbf{u} \tag{7}$$

which means that $\operatorname{cay}(\mathbf{u})$ is a quaternion. It is straightforward to verify that $\|\operatorname{cay}(\mathbf{u})\| = 1$, which means that $\operatorname{cay}(\mathbf{u})$ is a unit quaternion.

Let $\boldsymbol{\mu} = \tan\theta/4\mathbf{k}$ be the MRP [2] of the rotation given by θ and $\mathbf{k}$. It follows from (7) that $\operatorname{cay}(\boldsymbol{\mu}) = \mathbf{q}$, where $\mathbf{q} = \cos\theta/2 + \sin\theta/2\mathbf{k}$ is the corresponding unit quaternion. The inverse Cayley transform of $\mathbf{q}$ is given by

$$\operatorname{cay}^{-1}(\mathbf{q}) = (\mathbf{q}-1) \otimes (\mathbf{q}+1)^{-1}$$

and corresponds to the MRP representation of $\mathbf{q}$. This expression is undefined for $\mathbf{q} = -1$, which is the case when $\theta = 2\pi$, and is also the singularity of the MRP $\boldsymbol{\mu}$.

The Cayley transform of the dual vector $\tilde{\mathbf{u}} = \mathbf{u} + \varepsilon\mathbf{u}'$ is given by

$$\operatorname{cay}(\tilde{\mathbf{u}}) \triangleq (1+\tilde{\mathbf{u}}) \otimes (1-\tilde{\mathbf{u}})^{-1}. \tag{8}$$

This expression can be expanded using (3), and insertion of $(1-\mathbf{u})^{-1} = (1+\mathbf{u})/(1+\mathbf{u})^2$ gives

$$\operatorname{cay}(\tilde{\mathbf{u}}) = \operatorname{cay}(\mathbf{u}) + \frac{2\varepsilon(1+\mathbf{u}) \otimes \mathbf{u}' \otimes (1+\mathbf{u})}{(1+\|\mathbf{u}\|^2)^2}. \tag{9}$$

It follows from (2) that $\operatorname{cay}(\tilde{\mathbf{u}})$ is a unit dual quaternion.

The dual vector

$$\tilde{\boldsymbol{\mu}} = \tan\frac{\tilde{\theta}}{4}\tilde{\mathbf{k}} = \tan\frac{\theta}{4}\mathbf{k} + \varepsilon\left(\tan\frac{\theta}{4}\mathbf{k}' + \frac{d}{4\cos^2\frac{\theta}{4}}\mathbf{k}\right) \tag{10}$$

is a dual MRP and is used to relate the Cayley transform to the exponential function. Insertion of the real and dual parts of $\tilde{\boldsymbol{\mu}}$ into (9) gives $\operatorname{cay}(\tilde{\boldsymbol{\mu}}) = \tilde{\mathbf{q}}$ where $\tilde{\mathbf{q}}$ is given by (5). This means that

$$\operatorname{cay}\left(\tan\frac{\tilde{\theta}}{4}\tilde{\mathbf{k}}\right) = \exp\left(\frac{\tilde{\theta}\tilde{\mathbf{k}}}{2}\right). \tag{11}$$

The inverse Cayley transform of the unit dual quaternion $\tilde{\mathbf{q}}$ is given as

$$\operatorname{cay}^{-1}(\tilde{\mathbf{q}}) = (\tilde{\mathbf{q}}-1) \otimes (\tilde{\mathbf{q}}+1)^{-1} \tag{12}$$

and corresponds to the dual MRP representation of $\tilde{\mathbf{q}}$. This representation is called a twistor in [9]. The expression (12) can be written as

$$\operatorname{cay}^{-1}(\tilde{\mathbf{q}}) = \operatorname{cay}^{-1}(\mathbf{q}) + 2\varepsilon(\mathbf{q}+1)^{-1} \otimes \mathbf{q}' \otimes (\mathbf{q}+1)^{-1}$$

which is undefined for $\mathbf{q} = -1$. This means that the inverse Cayley transform of a unit dual quaternion is also undefined for $\theta = 2\pi$.

A detailed analysis of parameterizations of rigid body motions and the Cayley transform is presented in [26] for motors in conformal geometric algebra, which are isomorphic to unit dual quaternions.

E. Dual Quaternion Kinematics

Consider a body frame B and a reference inertial frame I. The displacement of B relative to I is described by the unit dual quaternion $\tilde{\mathbf{q}}_{B/I}$. The angular and linear velocities of B relative to I expressed in the coordinates of B are denoted $\boldsymbol{\omega}^B_{B/I}$ and $\mathbf{v}^B_{B/I}$, respectively. The twist of the body frame relative to the inertial frame described in the body frame is given by the dual vector $\tilde{\boldsymbol{\omega}}^B_{B/I} = \boldsymbol{\omega}^B_{B/I} + \varepsilon\mathbf{v}^B_{B/I}$. The kinematic differential equation describing the displacement of the system is given by

$$\dot{\tilde{\mathbf{q}}}_{B/I} = \frac{1}{2}\tilde{\mathbf{q}}_{B/I} \otimes \tilde{\boldsymbol{\omega}}^B_{B/I}. \tag{13}$$

For the remaining of this article, we will omit the subscripts B/I and B. This means that relevant entities describe the body B in relation to the inertial frame I expressed in the coordinates of B, unless otherwise stated.

Suppose that a displacement is given by a constant twist $\tilde{\boldsymbol{\omega}}$ in the time interval $\Delta t = t_{k+1} - t_k$. Then, the resulting displacement corresponds to a screw motion with $\tilde{\theta}\tilde{\mathbf{k}} = \Delta t\tilde{\boldsymbol{\omega}}$ and is given by the unit dual quaternion

$$\tilde{\mathbf{q}} = \exp\left(\frac{\Delta t\tilde{\boldsymbol{\omega}}_k}{2}\right).$$

This is used in the discrete-time integration of the unit dual quaternion. Suppose that the unit dual quaternion at time t_k is $\tilde{\mathbf{q}}_k$ and that the twist is constant and given by $\tilde{\boldsymbol{\omega}}_k$ in the time interval Δt. Then, the unit dual quaternion at time t_{k+1} is

$$\tilde{\mathbf{q}}_{k+1} = \tilde{\mathbf{q}}_k \otimes \exp\left(\frac{\Delta t\tilde{\boldsymbol{\omega}}_k}{2}\right). \tag{14}$$

Let $\tilde{\theta}\tilde{\mathbf{k}} = \Delta t\tilde{\boldsymbol{\omega}}_k$. From (11), it can be seen that

$$\operatorname{cay}\left(\tan\frac{\tilde{\theta}}{4}\tilde{\mathbf{k}}\right) = \exp\left(\frac{\Delta t\tilde{\boldsymbol{\omega}}_k}{2}\right).$$

Furthermore, from (10), it is seen that

$$\tan\frac{\tilde{\theta}}{4}\tilde{\mathbf{k}} \approx \frac{\theta}{4}\mathbf{k} + \varepsilon\left(\frac{\theta}{4}\mathbf{k}' + \frac{d\mathbf{k}}{4}\mathbf{k}\right) = \frac{\tilde{\theta}\tilde{\mathbf{k}}}{4}$$

when θ is small. This results in the approximation

$$\operatorname{cay}\left(\frac{\Delta t\tilde{\boldsymbol{\omega}}_k}{4}\right) \approx \exp\left(\frac{\Delta t\tilde{\boldsymbol{\omega}}_k}{2}\right), \quad \theta \text{ is small.} \tag{15}$$

This means that the Cayley transform can be used for the discrete-time integration in (14).

F. Particle Filter

Consider the discrete nonlinear dynamic system

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{w}_k), \quad \mathbf{w}_k \sim p_{\mathbf{w}_k}(\mathbf{w}_k) \tag{16}$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k), \ \ \mathbf{v}_k \sim p_{\mathbf{v}_k}(\mathbf{v}_k) \tag{17}$$

where $\mathbf{x}_k$ is the system state at time instance k and $\mathbf{z}_k$ is the measurement. The state transition model is described by the nonlinear function $f(\cdot)$ and the observation model is described by the nonlinear function $h(\cdot)$. The process noise $\mathbf{w}_k$ and the measurement noise $\mathbf{v}_k$ are assumed to be zero-mean white-noise sequences. The pdfs $p_{\mathbf{w}_k}(\mathbf{w}_k)$ and $p_{\mathbf{v}_k}(\mathbf{v}_k)$ are assumed to be known and mutually independent. The initial pdf of the state is also known and is given by $p_{\mathbf{x}_0}(\mathbf{x}_0)$.

The filtering is performed by calculating the conditional pdfs $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ recursively given the measurement sequence $\mathbf{z}_{1:k} = \mathbf{z}_1, \ldots, \mathbf{z}_k$. The prior pdf is given by

$$p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k})d\mathbf{x}_k \tag{18}$$

and the posterior pdf is given by

$$p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1}) = \frac{p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})}{\int p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})d\mathbf{x}_{k+1}}. \tag{19}$$

Analytic solutions to (18) and (19) are only available for a small and restrictive choice of systems and measurement models. A well-known solution is the Kalman filter, which assumes linear models and Gaussian noise with known covariances. The prior and posterior pdfs can then be represented by the mean and covariance.

The particle filter, more specifically the Bayesian bootstrap filter with roughening [11], approximates the pdfs by representing the distributions through discrete samples that are called particles. Each sample is a realization of the state vector and is given a weight.

1) Prediction: Given a set of particles, $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N$ at timestep k, where i is the particle number, w_k^i is the particle weight, and N is the number of particles. The prediction is performed by propagating each particle through (16)

$$\mathbf{x}_{k+1}^i = f(\mathbf{x}_k^i, \mathbf{w}_k^i), \quad \mathbf{w}_k^i \sim p_{\mathbf{w}_k}(\mathbf{w}_k)$$

where each sample of the process noise $\mathbf{w}_k^i$ is drawn from the pdf $p_{\mathbf{w}_k}(\mathbf{w}_k)$. This results in the set of predicted particles $\{\mathbf{x}_{k+1}^i, w_k^i\}_{i=1}^N$, where the weights are unchanged.

2) Update: The latest measurement $\mathbf{z}_{k+1}$ is accounted for by evaluating the likelihood $p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}^i)$ for each particle and updating the weights with

$$\bar{w}_{k+1}^i = w_k^i p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}^i) \tag{20}$$

and then normalizing

$$w_{k+1}^i = \frac{\bar{w}_{k+1}^i}{\sum_{i=1}^N \bar{w}_{k+1}^i}. \tag{21}$$

Note that an analytic expression for the likelihood $p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}^i)$ is required.

The estimates of a particle filter can be calculated from the posterior particles and weights. Estimates, such as the mean $\hat{\mathbf{x}}_{k+1}$ and covariance $\hat{\mathbf{P}}_{k+1}$, are found from

$$\hat{\mathbf{x}}_{k+1} \approx \sum_{i=1}^N w_{k+1}^i \mathbf{x}_{k+1}^i \tag{22}$$

and

$$\mathbf{P}_{k+1} \approx \sum_{i=1}^N w_{k+1}^i (\mathbf{x}_{k+1}^i - \hat{\mathbf{x}}_{k+1})(\mathbf{x}_{k+1}^i - \hat{\mathbf{x}}_{k+1})^\top. \tag{23}$$

3) Resampling and Roughening: The prediction and update step will cause the variance of the weights to increase, and over time most of the weight will be distributed over only a few particles [27]. The resampling step solves this problem by replacing particles in less probable regions with particles in probable regions, as first suggested in [11] and later justified in [28]. This can be done with systematic resampling, where samples are drawn (with replacement) N times from the set $\{\mathbf{x}_{k+1}^i, w_{k+1}^i\}_{i=1}^N$ to obtain the equally weighted particles $\{\mathbf{x}_{k+1}^i, 1/N\}_{i=1}^N$.

Resampling can cause the number of distinct particles to decrease, and therefore, a roughening step was suggested in [11]. Random jitter $\mathbf{c}_{k+1}^i \sim \mathcal{N}(0, \mathbf{J}_{k+1})$ is added to the resampled particles to increase the number of distinct particles in the significant region of the filter pdf. The roughened particles states $\{\mathbf{x}_{r,k+1}^i\}_{i=1}^N$ replace the resampled particle states and are calculated as

$$\mathbf{x}_{r,k+1}^i = \mathbf{x}_{k+1}^i + \mathbf{c}_{k+1}^i, \quad \mathbf{c}_{k+1}^i \sim \mathcal{N}(0, \mathbf{J}_{k+1}).$$

Here, $\mathbf{J}_{k+1}$ is a diagonal covariance matrix where the element m on the diagonal is calculated as

$$\mathbf{J}_{k+1}(m) = sM(m)N^{-1/n^x}$$

where n^x is the dimension of the state space, $M(m)$ is the maximum difference between component m in the state vectors $\mathbf{x}_{k+1}^i$ of the particles, and s is a tuning parameter.

The resampling and roughening steps are performed to achieve acceptable performance of the particle filter with a finite and practical number of particles. However, these steps are not required to run the filter and do not need to be applied at every cycle of the filter. To decide whether to resample and roughen or not, the number of effective samples N_{eff} [29] is approximated by

$$N_{\text{eff}} \approx \frac{1}{\sum_{i=1}^N (w_{k+1}^i)^2}. \tag{24}$$

If a few particles have significant weights, then $N_{\text{eff}} \approx 1$ or if all the particles have nearly equal weights, then $N_{\text{eff}} \approx N$.

III. Dual Quaternion Particle Filter

In this section, we develop a particle filer for pose estimation of a body B with respect to the inertial frame I. The particle filter is formulated with unit dual quaternions for global representation, while dual MRPs are used for local representation in the state. The discrete-time kinematics of the pose in terms of the dual quaternion is given by (14).

The prediction step of the particle filter is formulated to include angular and linear velocity measurements. This same formulation is used for no velocity measurements. This can be used if angular velocity measurements from a

body-mounted gyroscope are available, while linear velocity measurements are unavailable. The last formulation considers a body-mounted inertial measurement unit (IMU), providing angular velocity and specific force measurements.

In the update step, pose measurements in the form of unit dual quaternions are considered. There are no sensors that measure pose directly; however, a combination of sensors, such as GPS and a vision system, can be used to calculate equivalent dual quaternion measurements. More importantly, we use this to generate pose measurements for simulation and validation of the particle filter. In addition, point cloud measurements of the body from a 3-D camera are considered. We show how the state hypothesis of the particle filter can be weighted when we have a point cloud model of the body available.

Furthermore, we describe the special considerations in the resampling and roughening step of the particle filter when working with dual quaternions. A discussion is included on the challenges of run-time implementation and computational complexity of the algorithm.

A. Prediction

1) Angular and Linear Velocity Measurements: Angular velocity measurements $\boldsymbol{\omega}_{m,k}$ and linear velocity measurements $\mathbf{v}_{m,k}$ at discrete time instances t_k can be written on dual vector form as the measurement

$$\tilde{\boldsymbol{\omega}}_{m,k} = \boldsymbol{\omega}_{m,k} + \varepsilon \mathbf{v}_{m,k}.$$

If we assume that the velocity sensor is placed on the body and aligned with the body frame, we can apply a general measurement model on dual vector form [6] given by

$$\tilde{\boldsymbol{\omega}}_{m,k} = \tilde{\boldsymbol{\omega}}_k + \tilde{\mathbf{b}}_k + \tilde{\boldsymbol{\eta}}_{\omega,k} \tag{25}$$

where $\left[\tilde{\boldsymbol{\eta}}_{\omega,k}\right] \sim \mathcal{N}(0, \operatorname{diag}(\mathbf{Q}_\omega, \mathbf{Q}_\upsilon))$. Here, $\tilde{\boldsymbol{\eta}}_{\omega,k} = \boldsymbol{\eta}_{\omega,k} + \varepsilon \boldsymbol{\eta}_{\upsilon,k}$ is the dual vector measurement noise, while $\tilde{\mathbf{b}}_k = \mathbf{b}_{\omega,k} + \varepsilon \mathbf{b}_{\upsilon,k}$ is the dual vector bias of the measurement. The actual dual velocity $\tilde{\boldsymbol{\omega}}_k$ is then seen from (25) to be

$$\tilde{\boldsymbol{\omega}}_k = \tilde{\boldsymbol{\omega}}_{m,k} - \tilde{\mathbf{b}}_k - \tilde{\boldsymbol{\eta}}_{\omega,k}. \tag{26}$$

The model for the dual vector bias is given by the white noise process

$$\dot{\tilde{\mathbf{b}}} = \tilde{\boldsymbol{\eta}}_b$$

where the noise input $\tilde{\boldsymbol{\eta}}_b = \boldsymbol{\eta}_{b_\omega} + \varepsilon \boldsymbol{\eta}_{b_\upsilon}$ is a dual vector. This is discretized using the first-order Euler discretization

$$\tilde{\mathbf{b}}_{k+1} = \tilde{\mathbf{b}}_k + \Delta t \tilde{\boldsymbol{\eta}}_{b,k} \tag{27}$$

where $\left[\tilde{\boldsymbol{\eta}}_{b,k}\right] \sim \mathcal{N}\big(0, \operatorname{diag}\big(\mathbf{Q}_{b_\omega}, \mathbf{Q}_{b_\upsilon}\big)\big)$.

We now have a discrete system consisting of (14) and (27), where the dual velocity $\tilde{\boldsymbol{\omega}}_k$ in (14) is calculated from (26). We use the dual MRP as a local error representation of pose in the particle filter state vector, while the dual quaternion is used for global representation. The mapping between the local and global representation is performed with the Cayley transform

$$\delta\tilde{\mathbf{u}}_k = \operatorname{cay}^{-1}\big(\tilde{\mathbf{q}}_k \otimes \hat{\tilde{\mathbf{q}}}_k^*\big) \tag{28}$$

where $\delta\tilde{\mathbf{u}}_k$ is the dual MRP describing the deviation between the estimated pose $\hat{\tilde{\mathbf{q}}}_k$ and the actual pose $\tilde{\mathbf{q}}_k$. This approach can be seen as an extension of the method proposed in [10] for quaternions and is also analogous to the approach in [8] for dual quaternions. The particle filter state vector is given by

$$\mathbf{x}_k = \begin{bmatrix} \delta\tilde{\mathbf{u}}_k \\ \tilde{\mathbf{b}}_k \end{bmatrix} \tag{29}$$

giving a state dimension of $n_x = 12$.

The particle filter is initialized by drawing a set of N particle states $\mathbf{x}_0^i = [\delta\tilde{\mathbf{u}}_0^{i\top},\ \tilde{\mathbf{b}}_0^{i\top}]^\top$ from the initial distribution of the state $p_{\mathbf{x}_0}(\mathbf{x}_0)$. The initial posterior estimate of the pose is given by $\hat{\tilde{\mathbf{q}}}_0^+$. It is then assumed that the initial posterior pose deviation is $[\delta\hat{\tilde{\mathbf{u}}}_0^+] = [0,\ 0,\ 0,\ 0,\ 0,\ 0]^\top$. The initial posterior estimate of the dual bias is given by $\hat{\mathbf{b}}_0^+$. The weights are initialized to $w_0^i = 1/N$. It is seen from (28) that the initial value of the unit dual quaternion $\tilde{\mathbf{q}}_0^i$ of particle i can be obtained from

$$\tilde{\mathbf{q}}_0^i = \operatorname{cay}\big(\delta\tilde{\mathbf{u}}_0^i\big) \otimes \hat{\tilde{\mathbf{q}}}_0^+. \tag{30}$$

In the prediction, the unit dual quaternion $\tilde{\mathbf{q}}_k^i$ of particle i is propagated forward in time through

$$\tilde{\mathbf{q}}_{k+1}^i = \tilde{\mathbf{q}}_k^i \otimes \exp\left(\frac{\Delta t}{2}\tilde{\boldsymbol{\omega}}_k^i\right) \tag{31}$$

where the velocities are calculated as

$$\tilde{\boldsymbol{\omega}}_k^i = \tilde{\boldsymbol{\omega}}_{m,k} - \tilde{\mathbf{b}}_k^i - \tilde{\boldsymbol{\eta}}_{\omega,k}^i.$$

The noise vectors $\left[\tilde{\boldsymbol{\eta}}_{\omega,k}^i\right]$ are drawn from its pdf for each i. The bias states are propagated through

$$\tilde{\mathbf{b}}_{k+1}^i = \tilde{\mathbf{b}}_k^i + \Delta t \tilde{\boldsymbol{\eta}}_{b,k}^i \tag{32}$$

where the noise vectors $[\tilde{\boldsymbol{\eta}}_{b,k}^i]$ are drawn for every i.

It is also necessary to propagate the estimate of the pose according to

$$\hat{\tilde{\mathbf{q}}}_{k+1}^- = \hat{\tilde{\mathbf{q}}}_k^+ \otimes \exp\left(\frac{\Delta t}{2}\hat{\tilde{\boldsymbol{\omega}}}_k^+\right) \tag{33}$$

where

$$\hat{\tilde{\boldsymbol{\omega}}}_k^+ = \tilde{\boldsymbol{\omega}}_{m,k} - \hat{\tilde{\mathbf{b}}}_k^+.$$

The propagated state vector of particle i is then found from the inverse Cayley transform of the propagated unit dual quaternion $\tilde{\mathbf{q}}_{k+1}^i$ of particle i relative to the propagated estimate $\hat{\tilde{\mathbf{q}}}_k$. This gives

$$\mathbf{x}_{k+1}^i = \begin{bmatrix} \operatorname{cay}^{-1}\big(\tilde{\mathbf{q}}_{k+1}^i \otimes \hat{\tilde{\mathbf{q}}}_{k+1}^{-*}\big) \\ \tilde{\mathbf{b}}_{k+1}^i \end{bmatrix} = \begin{bmatrix} \delta\tilde{\mathbf{u}}_{k+1}^i \\ \tilde{\mathbf{b}}_{k+1}^i \end{bmatrix} \tag{34}$$

for $i = 1, \ldots, N$. The weights w_k^i are unchanged in the prediction.

2) No Velocity Measurements: When no velocity measurements are available, then $\tilde{\boldsymbol{\omega}}_{m,k}$ and $\tilde{\boldsymbol{\eta}}_{\omega,k}$ are set to zero, and (26) is reduced to

$$\tilde{\boldsymbol{\omega}}_k = -\tilde{\mathbf{b}}_k. \tag{35}$$

Consequently, the variances $\mathbf{Q}_\omega$ and $\mathbf{Q}_\upsilon$ of $\tilde{\boldsymbol{\eta}}_{\omega,k}$ are set to zero in the filter equations. It is then assumed that the velocities of the system follow a random walk process. As suggested

in [6], the bias state is used to describe the random walk velocity dynamics by simply increasing the variance of the bias $\tilde{\boldsymbol{\eta}}_{b,k}$ in the filter equations. This means that the filter remains exactly as in Section III-A1 when no velocity measurements are available, and only the tuning of the filter is different.

A similar approach can be used if only angular velocity measurements or linear velocity measurements are available. For example, if only angular velocity measurements from a gyroscope are available, then $\mathbf{Q}_\omega$ is set to the sensor noise and $\mathbf{Q}_{b_\omega}$ is the variance of the sensor bias, while $\mathbf{Q}_v$ is set to zero and $\mathbf{Q}_{b_v}$ is the variance of the random walk linear velocity.

3) Angular Velocity and Specific Force Measurements: In this case where angular velocity and specific force are measured, the real and dual parts of the velocity measurements must be handled separately. The dual velocity (26), written in terms of its real and dual parts, is reduced to

$$\boldsymbol{\omega} + \varepsilon \mathbf{v} = \left(\boldsymbol{\omega}_m - \mathbf{b}_\omega - \boldsymbol{\eta}_\omega\right) + \varepsilon(-\mathbf{b}_v) \tag{36}$$

as only angular velocity measurements are available. The dual part is written as

$$\mathbf{v} = -\mathbf{b}_v. \tag{37}$$

The linear acceleration of the body $\mathbf{a}^B_{B/I} = \mathbf{a}$ relative to the inertial frame can be related to the linear velocity by taking the derivative referenced to the inertial frame on both sides of (37)

$$\mathbf{a} = -\dot{\mathbf{b}}_v + \boldsymbol{\omega} \times \mathbf{v}. \tag{38}$$

Inserting (37) and the real part of (36) into (38) gives

$$\mathbf{a} = -\dot{\mathbf{b}}_v - \boldsymbol{\omega}_m \times \mathbf{b}_v + \mathbf{b}_\omega \times \mathbf{b}_v + \boldsymbol{\eta}_\omega \times \mathbf{b}_v. \tag{39}$$

A 3-D accelerometer measures the specific force $\mathbf{f}$, which is the acceleration minus the gravity acting on the sensor in the three dimensions of the Euclidean space. By placing the accelerometer in the body frame and aligning it with its axes, the specific force $\mathbf{f}^B_{B/I} = \mathbf{f}$ can be measured directly. The measurement model of the accelerometer is given by

$$\mathbf{f}_m = \mathbf{f} + \mathbf{b}_a + \boldsymbol{\eta}_a, \quad \boldsymbol{\eta}_a \sim \mathcal{N}(0, \mathbf{Q}_a) \tag{40}$$

where $\mathbf{f}_m$ is the measured specific force, $\mathbf{b}_a$ is the bias of the accelerometer, and $\boldsymbol{\eta}_a$ is a random component of zero-mean Gaussian noise. The bias is driven by a zero-mean white-noise process

$$\dot{\mathbf{b}}_a = \boldsymbol{\eta}_{b_a}$$

which is discretized using the first-order Euler discretization

$$\mathbf{b}_{a,k+1} = \mathbf{b}_{a,k} + \Delta t \boldsymbol{\eta}_{b_a,k}, \quad \boldsymbol{\eta}_{b_a,k} \sim \mathcal{N}\left(0, \mathbf{Q}_{b_a}\right). \tag{41}$$

The constant gravitational acceleration $\mathbf{g}^I$ is given in the inertial frame and is acting on the body. The acceleration of the body is then given as a sum of the specific force and the gravitational acceleration

$$\mathbf{a} = \mathbf{f} + \mathbf{q}^* \otimes \mathbf{g}^I \otimes \mathbf{q}. \tag{42}$$

Inserting (40) into (42) gives

$$\mathbf{a} = \mathbf{f}_m + \mathbf{q}^* \otimes \mathbf{g}^I \otimes \mathbf{q} - \mathbf{b}_a - \boldsymbol{\eta}_a. \tag{43}$$

Finally, inserting (39) into (43) and solving for $\dot{\mathbf{b}}_v$ gives the linear velocity bias dynamics

$$\dot{\mathbf{b}}_v = -\mathbf{f}_m - \boldsymbol{\omega}_m \times \mathbf{b}_v + \mathbf{b}_\omega \times \mathbf{b}_v + \boldsymbol{\eta}_\omega \times \mathbf{b}_v - \mathbf{q}^* \otimes \mathbf{g}^I \otimes \mathbf{q} + \mathbf{b}_a + \boldsymbol{\eta}_a.$$

This is again discretized using the first-order Euler discretization

$$\begin{aligned}\mathbf{b}_{v,k+1} = \mathbf{b}_{v,k} + \Delta t (&-\mathbf{f}_{m,k} - \boldsymbol{\omega}_{m,k} \times \mathbf{b}_{v,k} + \mathbf{b}_{\omega,k} \times \mathbf{b}_{v,k} \\ &+ \boldsymbol{\eta}_{\omega,k} \times \mathbf{b}_{v,k} - \mathbf{q}^*_k \otimes \mathbf{g}^I \otimes \mathbf{q}_k + \mathbf{b}_{a,k} + \boldsymbol{\eta}_{a,k}).\end{aligned} \tag{44}$$

We now expand the state vector to be

$$\mathbf{x}_k = \begin{bmatrix} \delta\tilde{\mathbf{u}}_k \\ \tilde{\mathbf{b}}_k \\ \mathbf{b}_{a,k} \end{bmatrix} = \begin{bmatrix} \delta\tilde{\mathbf{u}}_k \\ \mathbf{b}_{\omega,k} \\ \mathbf{b}_{v,k} \\ \mathbf{b}_{a,k} \end{bmatrix}. \tag{45}$$

The initialization of the particle filter is similar to that described in Section III-A1, except that the initial posterior estimate $\mathbf{b}^+_{a,k}$ is also given. The set of initial particle states $\{\mathbf{x}^i_0\}^N_{i=1} = \{[\delta\tilde{\mathbf{u}}^i_0 \ \tilde{\mathbf{b}}^i_0 \ \mathbf{b}^i_{a,0}]^\top\}^N_{i=1}$ is drawn from the initial distribution of the state $p_{x_0}(\mathbf{x}_0)$.

The particle filter prediction is performed as described in Section III-A1, except that the linear velocity bias $\mathbf{b}^i_{v,k}$ is now propagated through (44) as

$$\begin{aligned}\mathbf{b}^i_{v,k+1} = \mathbf{b}^i_{v,k} + \Delta t (&-\mathbf{f}_{m,k} - \boldsymbol{\omega}_{m,k} \times \mathbf{b}^i_{v,k} + \mathbf{b}^i_{\omega,k} \times \mathbf{b}^i_{v,k} \\ &+ \boldsymbol{\eta}^i_{\omega,k} \times \mathbf{b}^i_{v,k} - \mathbf{q}^{i*}_k \otimes \mathbf{g}^I \otimes \mathbf{q}^i_k + \mathbf{b}^i_{a,k} + \boldsymbol{\eta}^i_{a,k})\end{aligned} \tag{46}$$

instead of the dual part of (32), while the angular velocity bias $\mathbf{b}^i_{\omega,k}$ is still propagated through the real part of (32). In addition, the acceleration bias $\mathbf{b}^i_{a,k}$ is propagated through (41) as

$$\mathbf{b}^i_{a,k+1} = \mathbf{b}^i_{a,k} + \Delta t \boldsymbol{\eta}^i_{b_a,k}, \quad i = 1, \ldots, N. \tag{47}$$

B. Update

1) Dual Quaternion Measurements: The unit dual quaternion pose measurement $\tilde{\mathbf{q}}_{m,k+1}$ is now available. We use the measurement relation proposed in [17], where the noise $\left[\tilde{\boldsymbol{\eta}}_{q,k+1}\right] \sim \mathcal{N}(0, \mathbf{R})$ acting on the unit dual quaternion is modeled as a screw displacement. This is mapped with the Cayley transform to give the pose measurement model

$$\tilde{\mathbf{q}}_{m,k+1} = \tilde{\mathbf{q}}_{k+1} \otimes \mathrm{cay}\left(\frac{1}{2}\tilde{\boldsymbol{\eta}}_{q,k+1}\right). \tag{48}$$

The likelihood at time $t = k + 1$ of the pose measurement $\tilde{\mathbf{q}}_{m,k+1}$ given the particle pose $\tilde{\mathbf{q}}^i_{k+1}$ is denoted $p(\tilde{\mathbf{q}}_{m,k+1}|\tilde{\mathbf{q}}^i_{k+1})$. This is calculated as

$$p(\tilde{\mathbf{q}}_{m,k+1}|\tilde{\mathbf{q}}^i_{k+1}) \propto \exp\left(-\frac{1}{2}[\tilde{\boldsymbol{\eta}}^i_{q,k+1}]^\top \mathbf{R}^{-1}[\tilde{\boldsymbol{\eta}}^i_{q,k+1}]\right)$$

where

$$\tilde{\boldsymbol{\eta}}^i_{q,k+1} = 2\mathrm{cay}^{-1}\left(\tilde{\mathbf{q}}^{i*}_{k+1} \otimes \tilde{\mathbf{q}}_{m,k+1}\right).$$

The likelihoods $\{p(\tilde{\mathbf{q}}_{m,k+1}|\tilde{\mathbf{q}}^i_{k+1})\}^N_{i=1}$ are normalized before the weights are updated and normalized through (20) and (21).

2) Point Cloud Measurement: In this case, we want to estimate the pose of an arbitrary rigid object visible in a 3-D camera field of view. At each time step, a depth image is obtained from the 3-D camera, which can be represented as a set of measured points. More specifically, the set of measured points is termed a point cloud and is denoted by

$$\mathcal{P}_{k+1} = \{\mathbf{p}_{l,k+1}\}_{l=1}^{N_m}$$

where $\mathbf{p}_{l,k+1} \in \mathbb{R}^3$ and N_m is the number of points in $\mathcal{P}_{k+1}$.

The rigid object is represented by a model point cloud, which is denoted as

$$\mathcal{M} = \{\mathbf{m}_j\}_{j=1}^{N_\mathcal{M}}$$

where $\mathbf{m}_j \in \mathbb{R}^3$ and $N_\mathcal{M}$ is the number of points in $\mathcal{M}$. The model point cloud $\mathcal{M}$ can be obtained by sampling a CAD model, by scanning a 3-D image of the object, or it can be obtained by defining a cluster of points from an initial point cloud measurement.

Given a model $\mathcal{M}$ of the object and a set of prior particles $\{\mathbf{x}_{k+1}^i, \mathrm{w}_k^i\}_{i=1}^N$ with the corresponding unit dual quaternions $\{\tilde{\mathbf{q}}_{k+1}^i\}_{i=1}^N$ describing the pose of the object, we wish to update the particle weights given a measurement point cloud $\mathcal{P}_{k+1}$. The unit dual quaternion $\tilde{\mathbf{q}}_{k+1}^i$ of particle i is used to transform the points of the model $\mathcal{M}$ to the predicted pose as

$$\mathbf{m}_{j,k+1}^i = \mathbf{q}_{k+1}^i \otimes \mathbf{m}_j \otimes \mathbf{q}_{k+1}^{i*} + 2\mathbf{q}_{k+1}'^i \otimes \mathbf{q}_{k+1}^{i*}$$

for $j = 1, \ldots, N_\mathcal{M}$. This is done for all the particles resulting in a set of N models $\{\mathcal{M}_{k+1}^i\}_{i=1}^N = \{\{\mathbf{m}_{j,k+1}^i\}_{j=1}^{N_\mathcal{M}}\}_{i=1}^N$.

We use the point-to-point correspondence between the transformed points of the model $\mathcal{M}_{k+1}^i$ of particle i and the points in the measurement $\mathcal{P}_{k+1}$ to score how well particle i fits with the measurement. Two corresponding points are identified by calculating the nearest neighbors with replacement between the points in the measurement $\mathcal{P}_{k+1}$ and the points in $\mathcal{M}_{k+1}^i$. The nearest neighbors are obtained by calculating the Euclidean distance between all points

$$d(\mathbf{p}_l, \mathcal{M}_{k+1}^i) = \underset{\mathbf{m}_{j,k+1}^i \in \mathcal{M}_{k+1}^i}{\arg\min} \left\| \mathbf{m}_{j,k+1}^i - \mathbf{p}_{l,k+1} \right\|^2$$

for $l = 1, \ldots, N_m$. The set of correspondences is written as

$$C_{k+1}^i = \{(l, j) \mid \mathbf{p}_{l,k+1} \in \mathcal{P}_{k+1},\ \mathbf{m}_{j,k+1}^i \in \mathcal{M}_{k+1}^i\}.$$

When the nearest neighbor point correspondences between the measurement $\mathcal{P}_{k+1}$ and a model particle $\mathcal{M}_{k+1}^i$ have been established, a fitting score is calculated based on the Euclidean distance between the correspondences. The fitting score is calculated as

$$s_{k+1}^i = \sum_{C_{k+1}^i} \left\| \mathbf{m}_{j,k+1}^i - \mathbf{p}_{l,k+1} \right\|^2.$$

Following [13], the likelihood for each particle is then calculated from the fitting scores as

$$p(\mathcal{P}_{k+1}|\mathcal{M}_{k+1}^i) \propto \exp\left(-\lambda\left(1 - \frac{s_{k+1}^i - s_{\min,k+1}}{s_{\min,k+1} - s_{\max,k+1}}\right)\right) \quad (49)$$

where $s_{\min,k+1}$ and $s_{\max,k+1}$ are the minimal and maximal fitting score values for all model particles, respectively, and λ is a tuning scalar controlling the preference between higher and lower scoring particles. Evaluation of the exponential function in (49) will result in values, which lies in the interval (0, 1], where 1 corresponds to a perfect match. The likelihoods $\{p(\mathcal{P}_{k+1}|\mathcal{M}_{k+1}^i)\}_{i=1}^N$ are normalized before the weights are updated and normalized according to (20) and (21).

C. Resampling and Roughening

The number of effective particles N_{eff} is computed from (24) to decide whether to resample and roughen or not. These steps are performed exactly as described in II-F3. After resampling and roughening, we are left with a new set of particles $\{\mathbf{x}_{k+1}^i, 1/N\}_{i=1}^N$ representing the pose errors $\{\delta\tilde{\mathbf{u}}_{k+1}^i\}_{i=1}^N$ and the sensor biases. The predicted unit dual quaternions $\{\tilde{\mathbf{q}}_{k+1}^i\}_{i=1}^N$ then needs to be recalculated so that the resampling and roughening are accounted for. From (34), it can be seen that this can be done by displacing the propagated pose estimate with the resampled particle states as

$$\tilde{\mathbf{q}}_{k+1}^i = \mathrm{cay}\left(\delta\tilde{\mathbf{u}}_{k+1}^i\right) \otimes \hat{\tilde{\mathbf{q}}}_{k+1}^-. \quad (50)$$

The mean estimate $\hat{\mathbf{x}}_{k+1}^+ = [\delta\hat{\tilde{\mathbf{u}}}_{k+1}^+ \ \hat{\tilde{\mathbf{b}}}_{k+1}^+]^\top$ (or alternatively $\hat{\mathbf{x}}_{k+1}^+ = [\delta\hat{\tilde{\mathbf{u}}}_{k+1}^+ \ \hat{\tilde{\mathbf{b}}}_{k+1}^+ \ \hat{\mathbf{b}}_{a,k+1}^+]^\top$) and covariance estimate $\mathbf{P}_{k+1}^+$ are calculated before the resampling and roughening through (22) and (23), respectively. The resulting unit dual quaternion pose estimate is then obtained as

$$\hat{\tilde{\mathbf{q}}}_{k+1}^+ = \mathrm{cay}\left(\delta\hat{\tilde{\mathbf{u}}}_{k+1}^+\right) \otimes \hat{\tilde{\mathbf{q}}}_{k+1}^-. \quad (51)$$

Velocity estimates may be required in a control application and is found as the expectation of (26)

$$\hat{\tilde{\boldsymbol{\omega}}}_{k+1}^+ = \tilde{\boldsymbol{\omega}}_{m,k+1} - \hat{\tilde{\mathbf{b}}}_{k+1}^+. \quad (52)$$

Similarly, if the linear acceleration is required, it is found by taking the expectation of (43)

$$\hat{\mathbf{a}}_{k+1}^+ = \mathbf{f}_{m,k+1} + \hat{\mathbf{q}}_{k+1}^{+*} \otimes \mathbf{g}^I \otimes \hat{\mathbf{q}}_{k+1}^+ - \hat{\mathbf{b}}_{a,k+1}^+. \quad (53)$$

Note that the predicted mean $\hat{\mathbf{x}}_{k+1}^-$, covariance $\mathbf{P}_{k+1}^-$, and estimates $\hat{\tilde{\mathbf{q}}}_{k+1}^-$, $\hat{\tilde{\boldsymbol{\omega}}}_{k+1}^-$, $\hat{\mathbf{a}}_{k+1}^-$ can be obtained in a similar manner after the prediction step.

D. Implementation Considerations

The particle filter algorithm with N particles has a computational complexity of $\mathcal{O}(N)$. In addition, when considering point cloud measurements, the computational complexity increases depending on the number of model points $N_\mathcal{M}$ and the number of measured points N_m. The nearest neighbor search described in Section III-B2 will in a brute-force implementation result in a problem with $\mathcal{O}(N_m N_\mathcal{M} N)$ complexity that has to be computed for every iteration of the filter. A standard consumer-grade 3-D camera such as the Kinect v2 has a resolution of 424×512 pixels, which results in $N_m = 217\,088$ points in each measurement. In addition, we may want to run the filter with thousands of particles, making it impossible to meet any real-time requirements on the existing sequential hardware.

To accelerate the computation, we approximate the nearest neighbor search. This can be done by exploiting the structure

of the point clouds. However, only the measured point clouds from the 3-D camera inherit structure, whereas the model point cloud may be unstructured. Since we assume that the object is always in the field of view of the 3-D camera, we can expect that states with high likelihood will result in model point clouds $\mathcal{M}_{k+1}^i$ that are in the field of view. We can then project the model point clouds to the camera image plane, using the intrinsic camera calibration matrix K, and perform the NN-search in the depth image of the measured point cloud, similar to our previous work [30]. The NN-search is performed with a radius r around the pixel position of a point $\mathbf{m}_{j,k+1}^i \in \mathcal{M}_{k+1}^i$. The complexity now reduces to $\mathcal{O}(N_r N_{\mathcal{M}} N)$, where $N_r < N_m$ is the number of points that fall inside the radius r on the image plane.

Further acceleration of the computation can be achieved by exploiting the parallel structure of the particle filter. The prediction and update step for each particle can be performed in parallel on a graphical processing unit (GPU) such that the run-time complexity of the computation is reduced. GPU implementations of particle filters have lately become common practice to facilitate real-time estimation, and an example is [13]. A GPU consists of multiple processing cores, where the idea is to assign a particle to each processing core, which can be programmed using programming languages such as CUDA [31]. The parallelism we can achieve with GPUs strongly depends on the type of GPU, the number of particles, and the complexity of the calculations required for the particle filter. Generally, we can say that the run-time complexity of running a particle filter on a GPU is $\mathcal{O}(N_G)$, where $N_G = 1$ if the GPU is sufficiently powerful. In our case, a GPU implementation would result in a run-time complexity of $\mathcal{O}(N_r N_{\mathcal{M}} N_G)$. If limited resources are available or low run-time is required, both N and $N_{\mathcal{M}}$ can be tuned to achieve the tradeoff between run time and accuracy.

In our implementation of the particle filter, the prediction step, update step, and roughening step are performed on the GPU. The steps are programmed in CUDA C++ as modules with Python bindings. Python is used as a glue language controlling the execution sequence of the modules and loading and storing the data. In the prediction and roughening steps, random numbers are generated on the GPU using the cuRAND function in CUDA.

IV. Simulations

In this section, a simulation study is presented for the particle filter with unit dual quaternion measurements, as described in Section III-B1. Three different cases were simulated. In the first case, the filtering was performed solely based on pose measurements of a moving body. In the second case, angular velocity measurements from a body-mounted gyroscope were included. Here, the structure of the filter was the same as for the first case, and only the tuning of the process noise was changed. In the last case, both angular velocity and specific force measurements from a body-mounted IMU were used. The results of the particle filter were compared to the results of the twistor-based UKF (TUKF) [8] for all three simulation cases. The observation equation in the TUKF was replaced with (48) so that the observation equation was the same for both filters.

TABLE I
Initial States and Covariances for the Simulation With No Angular Velocity or Specific Force Measurements

Variable	Value
$[\hat{\tilde{\mathbf{q}}}_0^+]$	$[0.6549,\ 0.6634,\ -0.2048,\ -0.2986,\ 0,\ 0,\ 0,\ 0]^\top$
$\hat{\mathbf{x}}_0^+$	$[0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^\top$
$p_{x_0}(\mathbf{x}_0)$	$10^{-4}\mathbf{I}_{12}$
$\mathbf{R}$	$\mathrm{diag}(10^{-3}\mathbf{I}_3\mathrm{rad}^2,\ 8\times 10^{-3}\mathbf{I}_3\mathrm{m}^2)$
$\mathbf{Q}_\omega$	$10^{-9}\mathbf{I}_3(\mathrm{rad/s})^2$
$\mathbf{Q}_v$	$10^{-9}\mathbf{I}_3(\mathrm{m/s})^2$
$\mathbf{Q}_{b_\omega}$	$10^{-2}\mathbf{I}_3(\mathrm{rad/s^2})^2$
$\mathbf{Q}_{b_v}$	$10^{-2}\mathbf{I}_3(\mathrm{m/s^2})^2$

Ground-truth motion of a body B following random walk processes in linear and angular velocities was generated using the Python programming language and the NumPy module. The ground-truth velocities were generated with

$$\tilde{\boldsymbol{\omega}}_{k+1} = \tilde{\boldsymbol{\omega}}_k + \Delta t \tilde{\boldsymbol{\eta}}_{w,k},\ [\tilde{\boldsymbol{\eta}}_{w,k}] \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\mathbf{W}_\omega, \mathbf{W}_v))$$

where $\mathbf{W}_\omega = 10^{-2}\mathbf{I}_3(\mathrm{rad/s^2})^2$, $\mathbf{W}_v = 10^{-2}\mathbf{I}_3(\mathrm{m/s^2})^2$, $\boldsymbol{\omega}_0 = [0,\ 0,\ 0]^\top\mathrm{rad/s}$, $\mathbf{v}_0 = [0,\ 0,\ 0]^\top\mathrm{m/s}$, and $\Delta t = 0.01\,\mathrm{s}$. The ground-truth pose of the body was obtained from (14) using the ground-truth velocities and the initial pose $[\tilde{\mathbf{q}}_0] = [0.6549,\ 0.6634,\ -0.2048,\ -0.2986,\ 0,\ 0,\ 0,\ 0]^\top$, where the initial attitude was selected randomly. The dual quaternion pose measurements were then generated from (48) using $\mathbf{R} = \mathrm{diag}\big(10^{-3}\mathbf{I}_3(\mathrm{rad/s})^2,\ 8\times 10^{-3}\mathbf{I}_3(\mathrm{m/s})^2\big)$.

For all simulations cases, the prediction step of both the particle filter and TUKF was performed at a rate of 100 Hz, while the measurement update was at a rate of 5 Hz. A measure of the error in attitude between the true pose $\tilde{\mathbf{q}}_k$ and the filter estimate $\hat{\tilde{\mathbf{q}}}_k$ was calculated as $2\arccos(\eta_{\delta,k})$, where $\tilde{\mathbf{q}}_{\delta,k} = \hat{\tilde{\mathbf{q}}}_k^* \otimes \tilde{\mathbf{q}}_k = \eta_{\delta,k} + \boldsymbol{\sigma}_{\delta,k} + \varepsilon(\eta'_{\delta,k} + \boldsymbol{\sigma}'_{\delta,k})$. The magnitude of the error in position was calculated as $\|\mathbf{r}_k - \hat{\mathbf{r}}_k\|$, where $\mathbf{r}_k = 2\mathbf{q}'_k \otimes \mathbf{q}_k^*$ and $\hat{\mathbf{r}}_k = 2\hat{\mathbf{q}}'_k \otimes \hat{\mathbf{q}}_k^*$.

1) No Angular Velocity or Specific Force Measurements: Since only dual quaternion pose measurements were available, the prediction step of the particle filter was implemented as described in Section III-A2, giving a state dimension of $n_x = 12$. The tuning parameters $\mathbf{Q}_\omega$ and $\mathbf{Q}_v$ were set to a small value since no velocity measurements were available. The bias state of the filters had to account for the random walk velocity of the body, and therefore, $\mathbf{Q}_{b_\omega}$ and $\mathbf{Q}_{b_v}$ were set to the values of $\mathbf{W}_\omega$ and $\mathbf{W}_v$ used in the data generation of the ground-truth motion. The measurement noise $\mathbf{R}$ and initial states were set to the true values. The TUKF were tuned with the same values as the particle filter and the initial states and covariances for the filters are summarized in Table I.

The particle filter was tested with 100 Monte Carlo simulations for $N = 10\,000$ and $N = 50\,000$. The resampling threshold was set to $N_{\mathrm{eff}} = 0.5\,\mathrm{N}$, and the roughening tuning parameter was set to $s = 1 \times 10^{-5}$. The mean errors of the Monte Carlo simulations are plotted for $N = 50\,000$ in Fig. 1 with the corresponding errors for the TUKF. The root-mean-square (rms) errors after 60 s are shown in Table II. The results

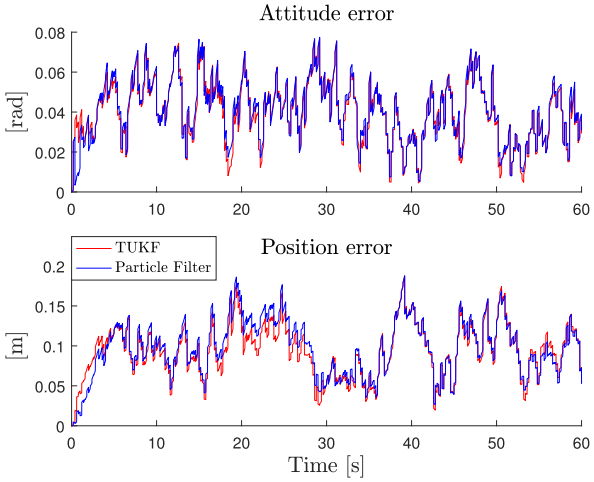

Fig. 1. Errors for the filters with no angular velocity or specific force measurements.

TABLE II
RMS ERRORS AFTER 60 s FOR THE SIMULATION WITH NO ANGULAR VELOCITY OR SPECIFIC FORCE MEASUREMENTS

	TUKF	Particle Filter	
N	-	10,000	50,000
Attitude $[10^{-2}\text{rad}]$	4.27	4.47	4.33
Position $[10^{-2}\text{m}]$	9.95	11.43	10.32

show that particle filter estimated the pose of the body with approximately the same accuracy as the TUKF. The particle filter approximates the true pdfs, where the approximation becomes more accurate when the number of particles N is increased. This is in agreement with the results of the simulations, where the rms error for $N = 50\,000$ was lower than for $N = 10\,000$.

2) Angular Velocity Measurements: In this case, we included angular velocity measurements in the prediction step of the particle filter and the TUKF. The angular velocity measurements were generated from the real part of (25) with $\mathbf{Q}_\omega = 10^{-4}\mathbf{I}_3(\text{rad/s})^2$, where the gyroscope bias was generated from the real part of (27) with $\mathbf{Q}_{b_\omega} = 5 \times 10^{-5}\mathbf{I}_3(\text{rad/s}^2)^2$ and $\mathbf{b}_{\omega,0} = [0,\ 0,\ 0]^\top\text{rad/s}$.

In the filters, the covariance of the angular velocity measurement noise was set to the true covariance used in the data generation, $\mathbf{Q}_\omega = 10^{-4}\mathbf{I}_3(\text{rad/s})^2$. The covariance of the bias was decreased to the true value $\mathbf{Q}_{b_\omega} = 5 \times 10^{-5}\mathbf{I}_3(\text{rad/s}^2)^2$, as the change in angular velocity was directly measured and no longer compensated for in the bias state. The rest of the filter parameters were the same as the parameters used in Section IV-1, as seen in Table I.

The 100 Monte Carlo simulations of the particle filter were tested for $N = 10\,000$ and $N = 50\,000$, with $N_{\text{eff}} = 0.5N$ and $s = 1 \times 10^{-5}$. The mean errors for $N = 50\,000$ are plotted in Fig. 2, along with the errors for the TUKF, and the rms errors after 60 s are shown in Table III. The particle filter and the TUKF estimated the pose of the body with comparable accuracy. Compared with the results in Section IV-1, the particle filter estimated the attitude with increased accuracy, while there was little difference in position accuracy.

3) Angular Velocity and Specific Force Measurements: In this case, both angular velocity measurements and specific

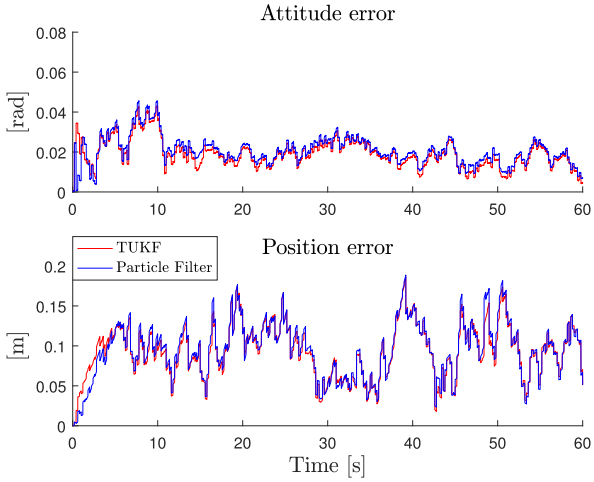

Fig. 2. Errors for the filters with angular velocity measurements.

TABLE III
RMS ERRORS AFTER 60 s FOR THE SIMULATION WITH ANGULAR VELOCITY MEASUREMENTS

	TUKF	Particle Filter	
N	-	10,000	50,000
Attitude $[10^{-2}\text{rad}]$	2.02	2.46	2.18
Position $[10^{-2}\text{m}]$	9.93	10.24	10.11

TABLE IV
INITIAL STATES AND COVARIANCES FOR THE SIMULATION WITH ANGULAR VELOCITY AND SPECIFIC FORCE MEASUREMENTS

Variable	Value
$[\hat{\check{\mathbf{q}}}_0^+]$	$[0.6549,\ 0.6634,\ -0.2048,\ -0.2986,\ 0,\ 0,\ 0,\ 0]^\top$
$\hat{\mathbf{x}}_0^+$	$[0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^\top$
$p_{x_0}(\mathbf{x}_0)$	$10^{-4}\mathbf{I}_{15}$
$\mathbf{R}$	$\text{diag}(10^{-3}\mathbf{I}_3\text{rad}^2,\ 8 \times 10^{-3}\mathbf{I}_3\text{m}^2)$
$\mathbf{Q}_\omega$	$10^{-4}\mathbf{I}_3(\text{rad/s})^2$
$\mathbf{Q}_a$	$10^{-4}\mathbf{I}_3(\text{m/s}^2)^2$
$\mathbf{Q}_{b_\omega}$	$5 \times 10^{-5}\mathbf{I}_3(\text{rad/s}^2)^2$
$\mathbf{Q}_{b_a}$	$5 \times 10^{-5}\mathbf{I}_3(\text{m/s}^3)^2$

force measurements were included in the prediction. To generate the specific force measurements, the acceleration of the body was obtained from the ground-truth motion as

$$\mathbf{a}_k = \frac{\mathbf{v}_{k+1} - \mathbf{v}_k}{\Delta t} + \boldsymbol{\omega}_k \times \mathbf{v}_k.$$

The specific force measurements from the accelerometer were then generated with $\mathbf{Q}_a = 2 \times 10^{-5}\mathbf{I}_3(\text{m/s}^2)^2$ from (43) by solving for $\mathbf{f}_m$ and setting $\mathbf{g}^I = [0,\ 0,\ -9.81]^\top\text{m/s}^2$. The accelerometer bias was generated from (41) with $\mathbf{Q}_{b_a} = 1.6 \times 10^{-8}\mathbf{I}_3(\text{m/s}^3)^2$ and $\mathbf{b}_{a,0} = [0,\ 0,\ 0]^\top\text{m/s}^2$.

The prediction of the particle filter was performed as described in Section III-A3 and the state vector was expanded according to (45), resulting in a state dimension of $n_x = 15$. The TUKF was implemented accordingly with angular velocity and specific force measurements. In this case, the tuning parameters of the process noise could be set to the true values used in the data generation. The initial states and covariances of the filters are summarized in Table IV. The number of effective particles was set to $N_{\text{eff}} = 0.5N$, and

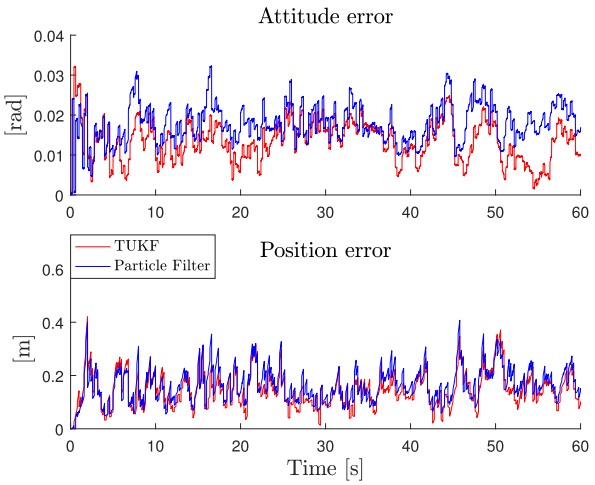

Fig. 3. Errors for the filters with angular velocity and specific force measurements.

TABLE V
RMS ERRORS AFTER 60 S FOR THE SIMULATION WITH ANGULAR VELOCITY AND SPECIFIC FORCE MEASUREMENTS

	TUKF	Particle Filter	
N	-	10,000	50,000
Attitude $[10^{-2}\text{rad}]$	1.44	-	1.85
Position $[10^{-2}\text{m}]$	15.94	-	17.57

the roughening tuning parameter was set to $s = 1\times10^{-4}$ for $N = 10\,000$ and $s = 5\times10^{-5}$ for $N = 50\,000$.

The mean errors of the 100 Monte Carlo simulations for $N = 50\,000$ are plotted in Fig. 3 together with the errors for the TUKF, and the rms errors after 60 s are shown in Table V. The particle filter estimated the pose of the body with no divergence runs for $N = 50\,000$. The accuracy of the TUKF was 22% better in attitude and 10% better in position. The 100 Monte Carlo simulations for $N = 10\,000$ resulted in 31 divergence runs.

A known limitation of particle filters is that the performance is strongly correlated with the dimension of the state [32]. As the dimension of the state space increases, the number of particles required to cover the state space quickly becomes unfeasible for particle applications. In this case, the high dimension of the state space in combination with small process noise resulted in many divergence runs for $N = 10\,000$. No divergence runs were observed when the number of particles was increased to $N = 50\,000$; however, the particle filter did not achieve the same accuracy as the TUKF. Marginalization [33], also known as Rao–Blackwellization, has been suggested to solve the problem of particle filtering in high-dimensional state spaces. The method utilizes the structure of the state-space model, where the particle filter is only used to filter the nonlinear states, while a KF is used to filter the linear states. Another approach is the feedback particle filter [34], where a feedback gain is used to adjust each particle according to the measurement, instead of calculating weights and resampling. Implementation of these methods is out of the scope of this article, and is left for future work.

V. EXPERIMENT

An experimental study was performed using the particle filter with point cloud measurements, as described in Section III-B2. In the experiment, the motion of a swinging payload hanging from a crane was estimated. Measurements of angular velocity or specific force were not available and the prediction step of the particle filter was calculated, as described in Section III-A2. This gave a state dimension of $n_x = 12$. The dynamics of the swinging payload could be modeled by the equations of motion of a spherical pendulum; however, the particle filter was modeled with a random walk velocity model and gave acceptable estimates of the motion by increasing the process noise.

TABLE VI
PARTICLE FILTER PARAMETERS FOR THE SWINGING PAYLOAD EXPERIMENT

Variable	N	N_{eff}	N_m	$N_{\mathcal{M}}$	λ	s
Value	10,000	5,000	217,088	3,138	10.0	1×10^{-5}

TABLE VII
INITIAL STATES AND COVARIANCES FOR THE SWINGING PAYLOAD EXPERIMENT

Variable	Value
$[\hat{\mathbf{q}}_0^+]$	$[1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^\top$
$\hat{\mathbf{x}}_0^+$	$[0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^\top$
$p_{x_0}(\mathbf{x}_0)$	$\text{diag}(5\mathbf{I}_3,\ 6.5\mathbf{I}_3,\ 5\mathbf{I}_3,\ 6.5\mathbf{I}_3)\times10^{-6}$
$\mathbf{Q}_\omega = \mathbf{Q}_v$	$0\mathbf{I}_3$
$\mathbf{Q}_{b_\omega}$	$5\mathbf{I}_3(\text{rad/s}^2)^2$
$\mathbf{Q}_{b_v}$	$6.5\mathbf{I}_3(\text{m/s}^2)^2$

Note that in this experiment, the likelihood of the particles (49) has an exponential distribution, which would have to be approximated by the Gaussian distribution in a Kalman filter, while the particle filter handles this optimally. In addition, the payload has to be detected from its surroundings in the point cloud measurements, in which the particle filter achieves by giving high weights to the best fitting hypothesis of the payload pose. The particle filter is also suited for this application as temporary occlusions and ambiguities in the point cloud measurements would result in multimodal pdfs. This was not studied in this experiment.

A. Experimental Setup

A Kinect v2 3-D camera was placed so that the swinging payload was visible in its field of view. The 3-D camera measured point clouds with a frequency of 30 Hz over an interval of 30 s. The model point cloud of the payload was identified from the first point cloud in the sequence by manually selecting points that represented the payload. The parameters of the particle filter for the experiment are given in Table VI, and the initial estimates and covariances are given in Table VII.

Fig. 4 shows snapshots from the pose estimation sequence after 10 s. In Fig. 4(a), the pose estimate of the particle filter is used to visualize the pose of the model point cloud for different time steps in the experiment. It can be seen that the model point cloud is projected onto the swinging payload, giving a visual indication of successful pose estimation.

B. Validation With Aruco Marker

To further validate the particle filter estimates, we used a separate pose estimation system consisting of an Aruco

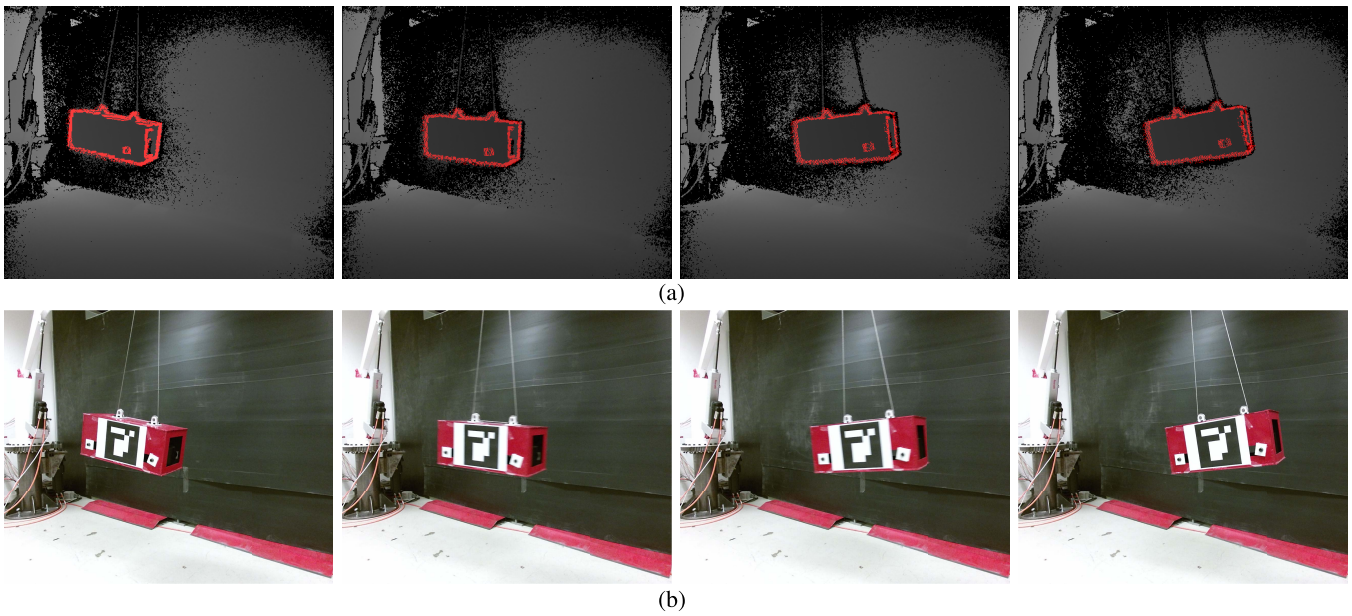

Fig. 4. Snapshots taken at $t = 10.0$ s, $t = 10.4$ s, $t = 10.8$ s, and $t = 11.2$ s from the payload pose estimation experiment. (a) Depth images (gray) from the Kinect v2 with the projected point cloud model $\mathcal{M}$ (red). (b) Corresponding 2-D images from the Kinect v2 RGB sensor showing the crane and payload with the Aruco marker.

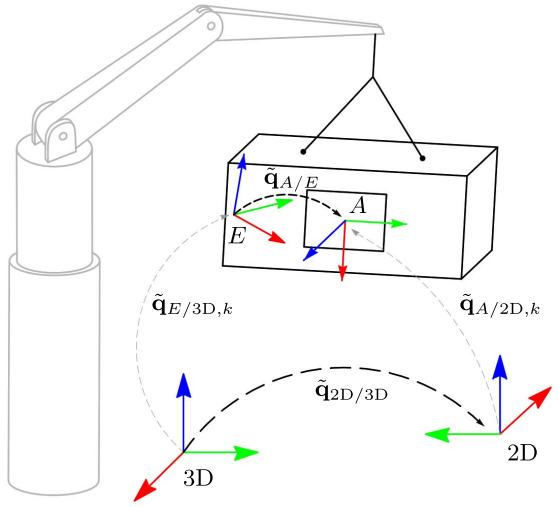

Fig. 5. Experimental setup of a payload hanging from a crane. The relevant frames and displacements are showing the kinematic chain described by (55).

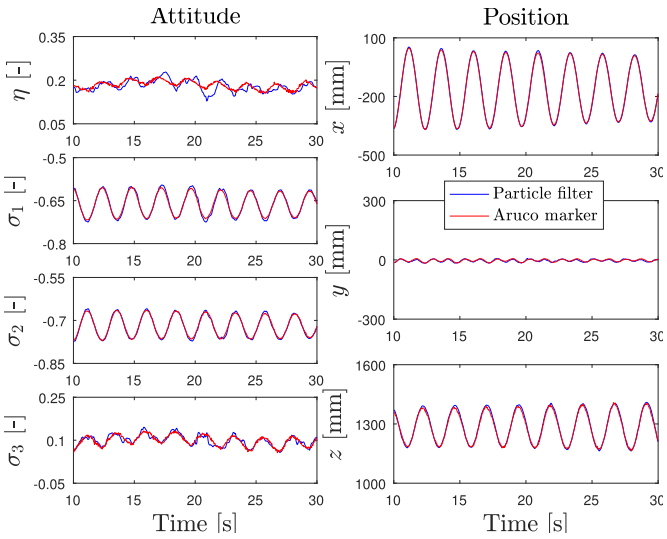

Fig. 6. Particle filter pose estimates of the swinging payload compared with the calibrated Aruco marker poses. Data for the first 10 s were used for calibration.

marker [35] and 2-D images from the Kinect v2 RGB sensor. The Aruco marker was attached to the payload [see Fig. 4(b)], and the 2-D images were obtained simultaneously with the 3-D point clouds during the pose estimation sequence. The pose of the Aruco marker was acquired using the Aruco module in OpenCV [36], which calculates the pose of the Aruco marker from the pixel positions of the detected corners using the intrinsic camera calibration matrix and the physical lengths of the sides of the Aruco marker.

The poses from the Aruco marker system described the displacement of the Aruco marker A relative to the 2-D camera frame and are denoted $\tilde{\mathbf{q}}_{A/2D,k}$. The particle filter pose estimates described the displacement of the payload relative to its initial pose. Thus, the poses from the Aruco marker system could not be directly compared with the particle filter pose estimates. To find the estimated motion of the payload relative to the 3-D camera frame, a frame E on the payload was manually selected from the first point cloud in the sequence and displaced with the particle filter pose estimates as

$$\tilde{\mathbf{q}}_{E/3D,k} = \hat{\tilde{\mathbf{q}}}_k^+ \otimes \tilde{\mathbf{q}}_{E/3D} \quad \forall k \tag{54}$$

where $\tilde{\mathbf{q}}_{E/3D}$ describes the pose of the frame E relative to the 3-D camera frame for $k = 1$. This results in the kinematic chain shown in Fig. 5, where $\tilde{\mathbf{q}}_{2D/3D}$ is the displacement of the 2-D camera frame relative to the 3-D camera frame and $\tilde{\mathbf{q}}_{A/E}$ is the displacement of the Aruco frame relative to the E frame. From Fig. 5, it can be seen that the particle filter pose

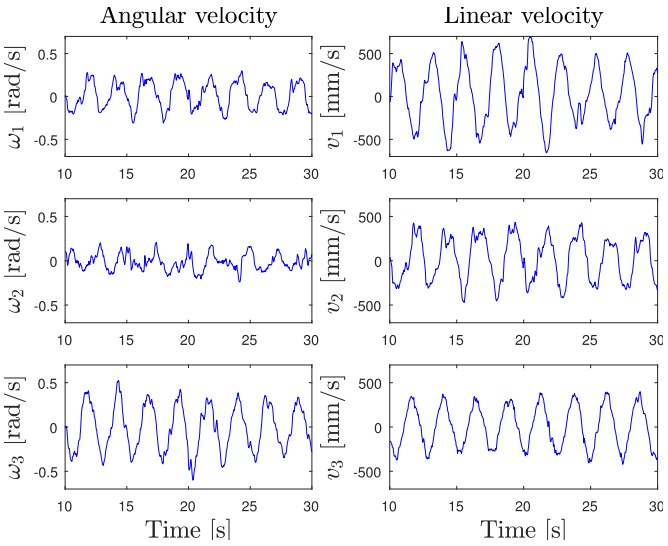

Fig. 7. Particle filter angular and linear velocity estimates of the swinging payload. The velocity estimates are extracted from the bias state of the particle filter through (52).

estimates are accurate if

$$\tilde{\mathbf{q}}_{E/3\mathrm{D},k} \approx \tilde{\mathbf{q}}_{2\mathrm{D}/3\mathrm{D}} \otimes \tilde{\mathbf{q}}_{A/2\mathrm{D},k} \otimes \tilde{\mathbf{q}}^*_{A/E} \quad \forall k. \tag{55}$$

The displacements $\tilde{\mathbf{q}}_{2\mathrm{D}/3\mathrm{D}}$ and $\tilde{\mathbf{q}}_{A/E}$ were obtained using the calibration procedure described in [24]. We experienced that it was imperative for the method that the displacement $\tilde{\mathbf{q}}_{A/E}$ was small, hence the necessity of (54).

The calibrated Aruco marker poses [right-hand side of (55)] are plotted in Fig. 6 together with the displaced particle filter pose estimates [left-hand side of (55)]. The positional components of the dual quaternions are converted into the Euclidean position for physical interpretation. Data from the first 10 s of the experiment were used for the calibration. The rms deviations after 30 s in attitude and position between the Aruco marker system and the particle filter were found to be 0.036 rad and 10.08 mm, respectively. The sinusoidal shapes of the curves in Fig. 6 clearly reflect the pendulum motion of the payload and the small deviations between the Aruco marker system and the particle filter indicate accurate pose estimation. The velocity estimates from the bias state of the particle filter are plotted in Fig. 7. It can be seen that the particle filter is able to estimate the oscillating velocities of the pendulum motion, even though the dynamics of the particle filter are modeled as random walk velocity.

VI. Conclusion

This article has presented a dual quaternion particle filter for pose estimation of a moving rigid body. The dual quaternion formulation can be seen as an extension of the previously proposed quaternion particle filter for attitude estimation. Two additional distinctions are the use of dual MRPs for pose representation instead of MRPs for attitude representation in the state vector and the use of an IMU for state propagation instead of a gyroscope only.

Previous publications have demonstrated the use of dual quaternions in Kalman filters, which require linearizations and Gaussian assumptions. Contrary to the Kalman filter approaches, the dual quaternion particle filter presented in this article can estimate the underlying probability distributions of nonlinear and non-Gaussian systems using multiple samples. As an example, we demonstrated how an exponential distribution could be used with the dual quaternion particle filter to estimate the pose of a swinging payload using a 3-D camera. From a theoretical point of view, a particle filter should also handle occlusions and multimodal distributions, even though this was not demonstrated in this article. The main limitation of the dual quaternion particle filter was the computational complexity caused by the high number of particles required to cover the state space.

The Cayley transform was used to perform a mapping between a unit dual quaternion and a dual MRP. This enabled propagation of the particles using the dual quaternion kinematic equation, while the dual MRPs were used in the state vector such that summation and weighting of the particle states could be performed without violating the algebraic constraints of the unit dual quaternion. This also reduced the dimension of the state vector since the dual quaternion has eight parameters, whereas the dual MRP has six.

The proposed particle filter was validated through Monte Carlo simulations using pose measurements in the form of unit dual quaternions with Gaussian noise. The particle filter estimated the pose of a moving body with similar accuracy as a UKF with dual quaternions when the state dimension was $n_x = 12$. This corresponded to the cases with and without angular velocity measurements in the prediction step. When specific force measurements were included in the prediction, the state dimension was increased to $n_x = 15$, resulting in decreased accuracy and robustness for the number of particles considered. Solutions to this problem were discussed.

Particle filters are suited for nonlinear systems with non-Gaussian noise. Therefore, visual measurements in the form of point clouds from a 3-D camera were also considered. An experimental study of the particle filter was performed, where the motion of a swinging payload was estimated using a 3-D camera. Fitting scores between a model point cloud of the payload and the point cloud measurement from the 3-D camera was calculated. The particle filter states were weighted with the likelihood of the fitting scores, which had an exponential probability distribution. The particle filter estimates were validated with a separate Aruco marker system that showed that the particle filter gave accurate results.

Acknowledgment

The authors would like to thank Dr. Torstein A. Myhre for proofreading this paper and sharing his CUDA C++ software implementation of the particle filter, which they extended and altered to fit their own algorithms.

References

[1] E. J. Leffens, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *J. Guid., Control, Dyn.*, vol. 5, no. 5, pp. 417–429, Sep. 1982.

[2] M. D. Shuster, "A survey of attitude representations," *Navigation*, vol. 8, no. 9, pp. 439–517, 1993.

[3] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *J. Guid., Control, Dyn.*, vol. 30, no. 1, pp. 12–28, Jan. 2007.

[4] E. Bayro-Corrochano and Y. Zhang, "The motor extended Kalman filter: A geometric approach for rigid motion estimation," *J. Math. Imag. Vis.*, vol. 13, no. 3, pp. 205–228, 2000.
[5] Y. Zu, U. Lee, and R. Dai, "Distributed motion estimation of space objects using dual quaternions," in *Proc. AIAA/AAS Astrodyn. Specialist Conf.*, Aug. 2014, p. 4296.
[6] N. Filipe, M. Kontitsis, and P. Tsiotras, "Extended Kalman filter for spacecraft pose estimation using dual quaternions," *J. Guid., Control, Dyn.*, vol. 38, no. 9, pp. 1625–1641, Sep. 2015.
[7] J. L. Crassidis and F. L. Markley, "Unscented filtering for spacecraft attitude estimation," *J. Guid., Control, Dyn.*, vol. 26, no. 4, pp. 536–542, Jul. 2003.
[8] Y. Deng, Z. Wang, and L. Liu, "Unscented Kalman filter for spacecraft pose estimation using twistors," *J. Guid., Control, Dyn.*, vol. 39, no. 8, pp. 1844–1856, Aug. 2016.
[9] Y. Deng and Z. Wang, "Modeling and control for spacecraft relative pose motion by using Twistor representation," *J. Guid., Control, Dyn.*, vol. 39, no. 5, pp. 1147–1154, May 2016.
[10] Y. Cheng and J. L. Crassidis, "Particle filtering for attitude estimation using a minimal local-error representation," *J. Guid., Control, Dyn.*, vol. 33, no. 4, pp. 1305–1310, Jul. 2010.
[11] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F, Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
[12] C. Choi and H. I. Christensen, "Robust 3D visual tracking using particle filtering on the special Euclidean group: A combined approach of keypoint and edge features," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 498–519, Apr. 2012.
[13] S. Li, S. Koo, and D. Lee, "Real-time and model-free object tracking using particle filter with joint color-spatial descriptor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 6079–6085.
[14] A. Blake and M. Isard, "The condensation algorithm-conditional density propagation and applications to visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 361–367.
[15] D. Titterton, J. L. Weston, and J. Weston, *Strapdown Inertial Navigation Technology*, vol. 17. Edison, NJ, USA: IET, 2004.
[16] J. M. Selig, "Exponential and Cayley maps for dual quaternions," *Adv. Appl. Clifford Algebras*, vol. 20, nos. 3–4, pp. 923–936, Oct. 2010.
[17] A. Sveier and O. Egeland, "Pose estimation using dual quaternions and moving horizon estimation," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 186–191, 2018.
[18] W. Blaschke, *Kinematics Quaternions*. Berlin, Germany: VEB Deutscher Verlag der Wissenshcaften, 1960.
[19] O. Egeland and J. T. Gravdahl, *Modeling and Simulation for Automatic Control*. Trondheim, Norway: Marine Cybernetics, 2002.
[20] F. C. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," *J. Mech. Des.*, vol. 117, no. 1, pp. 48–54, Mar. 1995.
[21] G. R. Veldkamp, "On the use of dual numbers, vectors and matrices in instantaneous, spatial kinematics," *Mechanism Mach. Theory*, vol. 11, no. 2, pp. 141–156, Jan. 1976.
[22] J. M. McCarthy and G. S. Soh, *Geometric Design Linkages*, vol. 11. New York, NY, USA: Springer, 2010, ch. 12.2, pp. 281–306.
[23] Y. Wu, X. Hu, D. Hu, T. Li, and J. Lian, "Strapdown inertial navigation system algorithms based on dual quaternions," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 110–132, Jan. 2005.
[24] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 286–298, Mar. 1999.
[25] A. M. Sjoberg and O. Egeland, "Kinematic feedback control using dual quaternions," in *Proc. 26th Medit. Conf. Control Autom. (MED)*, Jun. 2018, pp. 1–6.
[26] L. Tingelstad and O. Egeland, "Motor parameterization," *Adv. Appl. Clifford Algebras*, vol. 28, no. 2, p. 34, May 2018.
[27] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential monte carlo methods," in *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer, 2001, pp. 3–14.
[28] A. F. M. Smith and A. E. Gelfand, "Bayesian statistics without tears: A sampling–resampling perspective," *Amer. Statist.*, vol. 46, no. 2, pp. 84–88, May 1992.
[29] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.
[30] A. Sveier, T. A. Myhre, and O. Egeland, "Pose estimation with dual quaternions and iterative closest point," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 1913–1920.
[31] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with CUDA," in *Proc. ACM SIGGRAPH Classes*, 2008, p. 16.
[32] P. Bickel *et al.*, "Sharp failure rates for the bootstrap particle filter in high dimensions," in *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*. Institute of Mathematical Statistics, 2008, pp. 318–329.
[33] T. Schön, F. Gustafsson, and P.-J. Nordlund, *Marginalized Particle Filters for Nonlinear State-Space Models*. Linköping, Sweden: Univ. Electronic Press, 2003.
[34] K. Berntorp, "Feedback particle filter: Application and evaluation," in *Proc. 18th Int. Conf. Inf. Fusion (Fusion)*, Jul. 2015, pp. 1633–1640.
[35] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, Jun. 2014.
[36] OpenCV, "Open source computer vision library," 2018.

Aksel Sveier (Member, IEEE) received the M.Sc. and Ph.D. degrees in mechanical engineering from the Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology, Trondheim, Norway, in 2016 and 2020, respectively.

He is currently employed as a Research and Development Flight Control and Autonomy Engineer with FLIR Unmanned Aerial Systems, Hvalstad, Norway. His research interests include state estimation, navigation systems, computer vision, and modeling and control of robotic systems.

Olav Egeland (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in automatic control from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 1984 and 1987, respectively.

He was a Professor of robotics within electrical engineering at NTNU from 1989 to 2004. He was a co-founder of a start-up from 2004 to 2011. He is currently a Professor of production automation with the Department of Mechanical and Industrial Engineering, NTNU.

Dr. Egeland received the Automatica Prize Paper Award in 1996 and the IEEE TRANSACTIONS ON CONTROL SYSTEM TECHNOLOGY Outstanding Paper Award in 2000. He was an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 1996 to 1999 and *European Journal of Control* from 1998 to 2000. His research interest includes modeling and control for robotic and offshore applications.