Estimating $SE(3)$ elements using a dual quaternion based linear Kalman filter

Rangaprasad Arun Srivatsan, Gillian T. Rosen, D. Feroze Naina Mohamed and Howie Choset
Robotics Institute, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213.
Email: (arangapr,gtr,fdheenmo)@andrew.cmu.edu, choset@cs.cmu.edu

***Abstract*—Many applications in robotics such as registration, object tracking, sensor calibration, etc. use Kalman filters to estimate a time invariant $SE(3)$ element by locally linearizing a non-linear measurement model. Linearization-based filters tend to suffer from inaccurate estimates, and in some cases divergence, in the presence of large initialization errors. In this work, we use a dual quaternion to represent the $SE(3)$ element and use multiple measurements simultaneously to rewrite the measurement model in a truly linear form with state dependent measurement noise. Use of the linear measurement model bypasses the need for any linearization in prescribing the Kalman filter, resulting in accurate estimates while being less sensitive to initial estimation error. To show the broad applicability of this approach, we derive linear measurement models for applications that use either position measurements or pose measurements. A procedure to estimate the state dependent measurement uncertainty is also discussed. The efficacy of the formulation is illustrated using simulations and hardware experiments for two applications in robotics: rigid registration and sensor calibration.**

I. Introduction

Estimation of $SE(3)$ elements plays an important role in many applications of robotics, ranging from registration [11] and manipulation [18] to sensor calibration [9] and object tracking [31]. Probabilistic estimation techniques such as Kalman filters have been a popular choice for estimation of $SE(3)$ elements due to their ability to adapt to noisy sensor measurements. While the process models used for estimation of $SE(3)$ elements are linear, the update models are non-linear and the estimates provided by linear filters or their variants are often inaccurate and even diverge in cases where the initial estimation error is high. In this work, we derive a linear update model for estimation of $SE(3)$ using a dual quaternion representation.

Dual quaternions provide a means to combine both rotations and translations while retaining the advantages of using quaternions for representing rotations [6]. While dual-quaternions have been used with iterated extended Kalman filter (IEKF) to estimate $SE(3)$ elements, the update model was non-linear [10]. Non-linear update models can be highly sensitive to initial estimation errors, and can be computationally expensive. As a result in this work, we focus on deriving a linear update model to estimate $SE(3)$ elements. Chaukron et al [5] come closest to our work in terms of formulating a linear update model, but they only estimate the $SO(3)$ element. In this work, we start with a non-linear update model, and using multiple sensor measurements simultaneously, we rearrange the update model into a linear form. To the best of our knowledge, this is the first attempt to derive a linear update model for estimating time invariant $SE(3)$ elements using a Kalman filter.

The linear measurement model comes at the cost of state dependent measurement uncertainty. Measurement uncertainty is typically state independent and can be obtained based on the physical characteristics of the sensor and/or the measurement process. However, in case of state dependence, there is an additional burden of estimating the measurement uncertainty after each state update. State dependent measurement uncertainties have been used in systems for satellite tracking [27] and robot navigation [28]. We use an approach similar to [27], [5] to formulate the expressions for the state dependent measurement uncertainties. It should be noted that the measurement uncertainties have a linear dependence on their state vector, which allows for derivation of exact expressions of uncertainties [14].

In this work we look at two broad classes of applications based on the type of measurements used to estimate $SE(3)$ elements: 1) those that use position measurements such as registration from medical imaging [21], object tracking using laser range scanners [32], etc. and 2) those that use pose (position and orientation) measurements such as sensor calibration using inertial measurement units [19], hand-eye calibration using stereo vision [12], etc . The linear measurement models and state-dependent uncertainties are derived for both of these cases and the results are compared with non-linear filtering variants. We evaluate the formulation through simulations and experiments for two applications: registration and sensor calibration. We show that our dual quaternion-based linear filtering (DQF) produces more accurate and fast estimates even in the presence of high initial errors.

II. Background

Estimation of $SE(3)$ elements has been of interest for a long time in robotics literature. Horn et al [13] and Besl et al [3] developed methods for least squares estimation of $SE(3)$ elements for point registration. Park et al [23] and Chen et al [4] developed optimization based methods for estimating $SE(3)$ elements in sensor calibration problems. In the presence of noisy measurements, deterministic optimization methods have been observed to perform poorly [24]. However, probabilistic estimation techniques such as Kalman filters are effective at handling noisy measurements and producing accurate estimates of the state and associated uncertainty [15].

Several researchers have noted that filters used for estimating $SE(3)$ elements have non-linear update models [11], and hence linear Kalman filters produce poor estimates of $SE(3)$ elements. Several variants of the Kalman filter have been introduced to handle this non-linearity. The extended Kalman filter (EKF) and unscented Kalman filter (UKF) have been used to estimate $SE(3)$ elements for satellite orientation [27], manipulation [18], registration [24, 21] and sensor calibration [9]. EKF based filters perform first-order linear approximations of the non-linear update models and produce estimates which are known to diverge in the presence of high initial estimation errors [5]. UKF based methods do not linearize the models but instead require evaluation at multiple specially chosen points (called sigma points), which can be expensive for a high-dimensional system such as $SE(3)$. In addition UKF based methods require tuning of multiple parameters, which is not intuitive.

Prior work also has looked at several parameterizations of $SE(3)$ that would improve the performance of the filters. In [11] the state variables are confined over a known Riemannian manifold and a UKF is used to estimate the $SE(3)$ element. Lie algebra elements were used with an iterated extended Kalman filter (IEKF) in [29]. Both these methods involve highly non-linear update models with trigonometric terms in them. Quaternions are used to parametrize the rotation component of $SE(3)$ and an EKF is used to estimate the state in [20, 2]. Quaternions are used with a UKF in [17]. Quaternion representation-based filters usually involve a quadratic update model. Dual quaternions with an IEKF has been used in [10].

In this work, we use dual-quaternions to represent the $SE(3)$ element; using multiple simultaneous measurements, we derive a linear update model which can be used with a Kalman filter without the need for linearization.

III. Dual quaternions

There are many representations for $SE(3)$ elements such as Euler angles, quaternions, axis angles, etc. for rotation and Cartesian coordinates for translation. Dual quaternions compactly represent both translation and rotation, and with the methods presented in this paper, give rise to a linear update model. A detailed discussion on dual quaternions can be found in [16]. For the rest of this work, vectors will be described in bold font, quaternions will be represented by a ($\widetilde{}$) over a bold font and dual quaternion by a ($\hat{}$) over a bold font.

A. Quaternions

A quaternion $\widetilde{\boldsymbol{q}}$ is a 4-tuple (q_0, q_1, q_2, q_3), where q_0 is the scalar part and $\boldsymbol{q} = (q_1, q_2, q_3)^T = \text{vec}(\widetilde{\boldsymbol{q}})$ is the vector part of the quaternion. A 3-vector can be denoted by a quaternion with a 0 scalar part.

Multiplication of two quaternions $\widetilde{\boldsymbol{p}}$ and $\widetilde{\boldsymbol{q}}$ is given by

$$\widetilde{\boldsymbol{p}} \odot \widetilde{\boldsymbol{q}} = p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q} + q_0 \boldsymbol{p} + p_0 \boldsymbol{q} + \boldsymbol{p} \times \boldsymbol{q}, \tag{1}$$

$$= \begin{bmatrix} p_0 & -\boldsymbol{p}^T \\ \boldsymbol{p} & \boldsymbol{p}^\times + p_0 \boldsymbol{I}_3 \end{bmatrix} \widetilde{\boldsymbol{q}} = \begin{bmatrix} q_0 & -\boldsymbol{q}^T \\ \boldsymbol{q} & -\boldsymbol{q}^\times + q_0 \boldsymbol{I}_3 \end{bmatrix} \widetilde{\boldsymbol{p}} \tag{2}$$

where $\odot$ is the quaternion multiplication operator and $[\boldsymbol{v}]^\times$ is the skew-symmetric matrix formed from the vector $\boldsymbol{v}$. Given a quaternion $\widetilde{\boldsymbol{q}}$, its conjugate $\widetilde{\boldsymbol{q}}^*$ can be written as: $\widetilde{\boldsymbol{q}}^* = (q_0, -q_1, -q_2, -q_3)$. If the scalar part of a quaternion is 0,

$$\widetilde{\boldsymbol{q}}^* = -\widetilde{\boldsymbol{q}}^*. \tag{3}$$

The conjugate has the following property: $\text{vec}\left(\widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{q}}^*\right) = 0$. The norm of a quaternion is $|\widetilde{\boldsymbol{q}}| = \sqrt{\text{scalar}(\widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{q}}^*)}$ and a unit quaternion is one with $|\widetilde{\boldsymbol{q}}| = 1$. Unit quaternions can be used to represent rotation about an axis (denoted by the unit vector $\boldsymbol{k}$) by an angle $\theta \in [-\pi, \pi]$ as follows

$$\widetilde{\boldsymbol{q}} = \cos\left(\frac{\theta}{2}\right) + \boldsymbol{k} \sin\left(\frac{\theta}{2}\right). \tag{4}$$

B. Dual quaternions

A dual quaternion $\hat{\boldsymbol{x}}$ is an 8-tuple $(p_0, p_1, p_2, p_3, q_0, q_1, q_2, q_3)$, which can be written in the form: $\hat{\boldsymbol{x}} = \widetilde{\boldsymbol{p}} + \epsilon \widetilde{\boldsymbol{q}}$, where $\widetilde{\boldsymbol{p}} = (p_0, p_1, p_2, p_3)$, $\widetilde{\boldsymbol{q}} = (q_0, q_1, q_2, q_3)$, and ϵ is an element having the following property: $\epsilon \neq 0 \quad \text{and } \epsilon^2 = 0$. ϵ is a mathematical construct with a defined property and is not to be confused as having a small value close to 0. $\widetilde{\boldsymbol{p}}$ is called the real part and $\widetilde{\boldsymbol{q}}$ is called the dual part of the dual quaternion.

Multiplication of two dual quaternions $\hat{\boldsymbol{x}}_1 = \widetilde{\boldsymbol{p}}_1 + \epsilon \widetilde{\boldsymbol{q}}_1$ and $\hat{\boldsymbol{x}}_2 = \widetilde{\boldsymbol{p}}_2 + \epsilon \widetilde{\boldsymbol{q}}_2$ is given as

$$\hat{\boldsymbol{x}}_1 \otimes \hat{\boldsymbol{x}}_2 = \widetilde{\boldsymbol{p}}_1 \odot \widetilde{\boldsymbol{p}}_2 + \epsilon \left(\widetilde{\boldsymbol{p}}_1 \odot \widetilde{\boldsymbol{q}}_2 + \widetilde{\boldsymbol{q}}_1 \odot \widetilde{\boldsymbol{p}}_2\right), \tag{5}$$

where $\otimes$ is the dual quaternion multiplication operator.

Dual quaternions have three conjugates: 1) First conjugate: $\hat{\boldsymbol{x}}^{1*} = \widetilde{\boldsymbol{p}} - \epsilon \widetilde{\boldsymbol{q}}$, 2) Second conjugate: $\hat{\boldsymbol{x}}^{2*} = \widetilde{\boldsymbol{p}}^* + \epsilon \widetilde{\boldsymbol{q}}^*$, and 3) Third conjugate: $\hat{\boldsymbol{x}}^{3*} = \widetilde{\boldsymbol{p}}^* - \epsilon \widetilde{\boldsymbol{q}}^*$. A dual quaternion is called "unit" if $\hat{\boldsymbol{x}} \otimes \hat{\boldsymbol{x}}^{2*} = 1$. An important property of the third conjugate that will be used in this work is, $(\hat{\boldsymbol{x}}_1 \otimes \hat{\boldsymbol{x}}_2)^{3*} = \hat{\boldsymbol{x}}_2^{3*} \otimes \hat{\boldsymbol{x}}_1^{3*}$.

A dual quaternion used to represent a vector $\boldsymbol{a} \in \mathbb{R}^3$ has the following form

$$\hat{\boldsymbol{a}} = 1 + \epsilon\left(\widetilde{\boldsymbol{a}}\right), \quad \text{where } \widetilde{\boldsymbol{a}} = 0 + \boldsymbol{a}. \tag{6}$$

A dual quaternion that is used to represent an $SE(3)$ element has the following form

$$\hat{\boldsymbol{x}} = \widetilde{\boldsymbol{q}}_r + \epsilon \frac{\widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r}{2}, \tag{7}$$

where $\widetilde{\boldsymbol{q}}_r$ is the rotation quaternion whose form is as shown in Eq. 4 and $\widetilde{\boldsymbol{q}}_t = 0 + \boldsymbol{t}$ is the quaternion representation of the translational component of the $SE(3)$ element, $\boldsymbol{t} \in \mathbb{R}^3$. For the sake of simplicity, we rewrite Eq. 7 as

$$\hat{\boldsymbol{x}} = \widetilde{\boldsymbol{q}}_r + \epsilon \widetilde{\boldsymbol{q}}_d, \quad \text{where} \tag{8}$$

$$\widetilde{\boldsymbol{q}}_d = \frac{\widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r}{2}. \tag{9}$$

It is important to note that $\hat{\boldsymbol{x}}$ is a unit dual quaternion since its dual-product with the second conjugate is unity.

Let point $\boldsymbol{a} \in \mathbb{R}^3$ be obtained by transforming point $\boldsymbol{b} \in \mathbb{R}^3$ using a dual quaternion $\hat{\boldsymbol{x}}$. The transformation can be mathematically described as

$$\hat{\boldsymbol{a}} = \hat{\boldsymbol{x}} \otimes \hat{\boldsymbol{b}} \otimes \hat{\boldsymbol{x}}^{3*}, \tag{10}$$

where $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{b}}$ are obtained using Eq. 6.

Lemma III.1. *For a unit dual quaternion, $\hat{\boldsymbol{x}} = \widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d$, the product of third and first conjugate equals unity: $\hat{\boldsymbol{x}}^{3*} \otimes \hat{\boldsymbol{x}}^{1*} = 1$.*
Proof:

$$\begin{aligned}\hat{\boldsymbol{x}}^{3*} \otimes \hat{\boldsymbol{x}}^{1*} &= (\widetilde{\boldsymbol{q}}_r^* - \epsilon\widetilde{\boldsymbol{q}}_d^*) \otimes (\widetilde{\boldsymbol{q}}_r - \epsilon\widetilde{\boldsymbol{q}}_d) \\ &= \widetilde{\boldsymbol{q}}_r^* \odot \widetilde{\boldsymbol{q}}_r - \epsilon\left(\widetilde{\boldsymbol{q}}_r^* \odot \widetilde{\boldsymbol{q}}_d + \widetilde{\boldsymbol{q}}_d^* \odot \widetilde{\boldsymbol{q}}_r\right), \textit{ from Eq. 1} \\ &= 1 - \epsilon\left(\widetilde{\boldsymbol{q}}_r^* \odot \frac{\widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r}{2} + \frac{\widetilde{\boldsymbol{q}}_r^* \odot \widetilde{\boldsymbol{q}}_t^*}{2} \odot \widetilde{\boldsymbol{q}}_r\right).\end{aligned} \tag{11}$$

Using the property that $\widetilde{\boldsymbol{q}}_r$ is a unit quaternion and $\widetilde{\boldsymbol{q}}_t^* = -\widetilde{\boldsymbol{q}}_t$ from Eq. 3. Eq. 11 can be further simplified as $\hat{\boldsymbol{x}}^{3*} \otimes \hat{\boldsymbol{x}}^{1*} = 1$.

IV. Mathematical modeling

Most applications that estimate a time invariant $SE(3)$ element can be broadly divided into two cases: Case I, ones that use position measurements and Case II, that use pose measurements for updating the state. The measurement model for both these cases are non-linear and algebraically very different. Dual quaternions provide us the means to rewrite the measurement models for both these cases in a linear form. The rest of the section deals with the derivation of measurement models for the two cases and the corresponding uncertainties.

A. Measurement model for Case I

Systems that use position-measurements for model update have the following general form

$$\boldsymbol{a} = \boldsymbol{R}\boldsymbol{b} + \boldsymbol{t}, \tag{12}$$

where $\boldsymbol{a}$ is the sensor measurement, $\boldsymbol{R} \in SO(3)$ is the rotation matrix, $\boldsymbol{b} \in \mathbb{R}^3$ is the point to be transformed and $\boldsymbol{t} \in \mathbb{R}^3$ is the translation vector. In an application such as rigid registration of images, $\boldsymbol{a}$ is the sensed location of points and $\boldsymbol{b}$ is the corresponding point on the CAD model of the object. Eq. 12 can be rewritten using dual-quaternions from Eq. 10 as shown

$$\hat{\boldsymbol{a}} = \hat{\boldsymbol{x}} \otimes \hat{\boldsymbol{b}} \otimes \hat{\boldsymbol{x}}^{3*}, \tag{13}$$

where $\hat{\boldsymbol{x}}$ is as defined in Eq. 8. Applying Lemma III.1, Eq. 13 can be rewritten as

$$\hat{\boldsymbol{a}} \otimes \hat{\boldsymbol{x}}^{1*} = \hat{\boldsymbol{x}} \otimes \hat{\boldsymbol{b}}. \tag{14}$$

Let us consider the case of a pair of measurements $\boldsymbol{a}_i, i = 1, 2$. From Eq. 14, we have

$$\begin{aligned}&\hat{\boldsymbol{a}}_i \otimes \hat{\boldsymbol{x}}^{1*} = \hat{\boldsymbol{x}} \otimes \hat{\boldsymbol{b}}_i, \\ \Rightarrow &(1 + \epsilon\widetilde{\boldsymbol{a}}_1) \otimes (\widetilde{\boldsymbol{q}}_r - \epsilon\widetilde{\boldsymbol{q}}_d) = (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) \otimes (1 + \epsilon\widetilde{\boldsymbol{b}}_1), \quad (15) \\ &(1 + \epsilon\widetilde{\boldsymbol{a}}_2) \otimes (\widetilde{\boldsymbol{q}}_r - \epsilon\widetilde{\boldsymbol{q}}_d) = (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) \otimes (1 + \epsilon\widetilde{\boldsymbol{b}}_2) \quad (16)\end{aligned}$$

Subtracting Eq. 16 from Eq. 15, we obtain

$$\begin{aligned}&(\epsilon(\widetilde{\boldsymbol{a}}_1 - \widetilde{\boldsymbol{a}}_2)) \otimes (\widetilde{\boldsymbol{q}}_r - \epsilon\widetilde{\boldsymbol{q}}_d) = (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) \otimes \left(\epsilon\left(\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2\right)\right) \\ &\Rightarrow \epsilon((\widetilde{\boldsymbol{a}}_1 - \widetilde{\boldsymbol{a}}_2) \odot \widetilde{\boldsymbol{q}}_r) = \epsilon\left(\widetilde{\boldsymbol{q}}_r \odot \left(\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2\right)\right) \\ &\Rightarrow (\widetilde{\boldsymbol{a}}_1 - \widetilde{\boldsymbol{a}}_2) \odot \widetilde{\boldsymbol{q}}_r - \widetilde{\boldsymbol{q}}_r \odot \left(\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2\right) = \tilde{\boldsymbol{0}}.\end{aligned} \tag{17}$$

Note that Eq. 17 does not have $\widetilde{\boldsymbol{q}}_d$ and contains only the rotation quaternion. Using the quaternion multiplication described in Eq. 1, Eq. 17 can be rewritten in the following form

$$\boldsymbol{H}\widetilde{\boldsymbol{q}}_r = \tilde{\boldsymbol{0}}, \quad \text{where} \tag{18}$$

$$\boldsymbol{H} = \begin{bmatrix} 0 & -(\boldsymbol{a}_1 - \boldsymbol{a}_2 - \boldsymbol{b}_1 + \boldsymbol{b}_2)^T \\ (\boldsymbol{a}_1 - \boldsymbol{a}_2 - \boldsymbol{b}_1 + \boldsymbol{b}_2) & (\boldsymbol{a}_1 + \boldsymbol{a}_2 + \boldsymbol{b}_1 + \boldsymbol{b}_2)^\times \end{bmatrix}_{4\times4}. \tag{19}$$

The rotation quaternion $\widetilde{\boldsymbol{q}}_r$ lies in the null space of $\boldsymbol{H}$. In order to estimate $\widetilde{\boldsymbol{q}}_r$ we use a Kalman filter whose state vector is $\widetilde{\boldsymbol{q}}_r$. For this filter, the pseudo measurement model is

$$\boldsymbol{h} = \boldsymbol{H}\widetilde{\boldsymbol{q}}_r. \tag{20}$$

We enforce the pseudo measurement model $\boldsymbol{h} = \boldsymbol{0} \in \mathbb{R}^4$. The measurement in Eq. 20 is called "pseudo-measurement" because the $\boldsymbol{h}$ does not represent a true measurement. The pseudo-measurement is dependent on $\widetilde{\boldsymbol{q}}_r$, $\widetilde{\boldsymbol{a}}_i$ and $\widetilde{\boldsymbol{b}}_i$ all of which have associated uncertainties. In section IV-C1, we discuss the procedure to compute the uncertainty in the pseudo-measurement. Subsequently in section V, we describe the equations of the Kalman filter that estimates $\widetilde{\boldsymbol{q}}_r$ using the linear measurement model.

After estimating $\widetilde{\boldsymbol{q}}_r$ using a Kalman filter, we need to estimate $\widetilde{\boldsymbol{q}}_t$. Adding the Eq. 15 and Eq. 16 we have

$$\begin{aligned}&(2 + \epsilon(\widetilde{\boldsymbol{a}}_1 + \widetilde{\boldsymbol{a}}_2)) \otimes (\widetilde{\boldsymbol{q}}_r - \epsilon\widetilde{\boldsymbol{q}}_d) = \\ &\qquad (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) \otimes \left(2 + \epsilon(\widetilde{\boldsymbol{b}}_1 + \widetilde{\boldsymbol{b}}_2)\right), \\ &\Rightarrow 2\widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r = (\widetilde{\boldsymbol{a}}_1 + \widetilde{\boldsymbol{a}}_2) \odot \widetilde{\boldsymbol{q}}_r - \widetilde{\boldsymbol{q}}_r \odot (\widetilde{\boldsymbol{b}}_1 + \widetilde{\boldsymbol{b}}_2), \\ &\Rightarrow \widetilde{\boldsymbol{q}}_t = \frac{\widetilde{\boldsymbol{a}}_1 + \widetilde{\boldsymbol{a}}_2}{2} - \widetilde{\boldsymbol{q}}_r \odot \frac{\widetilde{\boldsymbol{b}}_1 + \widetilde{\boldsymbol{b}}_2}{2} \odot \widetilde{\boldsymbol{q}}_r^*.\end{aligned} \tag{21}$$

Thus, Eq. 21 determines $\widetilde{\boldsymbol{q}}_t$ directly using the estimated value of $\widetilde{\boldsymbol{q}}_r$ without the need for a Kalman filter. This is a helpful byproduct of using multiple measurements simultaneously in Eq. 15. Since the scalar part of $\widetilde{\boldsymbol{q}}_t$ is 0 and vector part is $\boldsymbol{t}$, we can rewrite Eq. 21 in the following vector form

$$\boldsymbol{t} = \frac{\boldsymbol{a}_1 + \boldsymbol{a}_2}{2} - \boldsymbol{R}_{\widetilde{\boldsymbol{q}}_r}\left(\frac{\boldsymbol{b}_1 + \boldsymbol{b}_2}{2}\right), \tag{22}$$

where $\boldsymbol{R}_{\widetilde{\boldsymbol{q}}_r}$ is the rotation matrix formed using the quaternion $\widetilde{\boldsymbol{q}}_r$. Section IV-C2 describes the uncertainty associated with $\boldsymbol{t}$.

B. Measurement model for Case II

Systems that use pose-measurements for model update typically have the following general form [23]

$$\boldsymbol{A}\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B} = \boldsymbol{0}, \tag{23}$$

where $\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{B} \in SE(3)$. $\boldsymbol{A}$ and $\boldsymbol{B}$ are pose-measurements and $\boldsymbol{X}$ is the desired transformation to be estimated.

A Kalman filter used to estimate $\boldsymbol{X}$ such as in [9], would have a pseudo-measurement of the form, $\boldsymbol{h}_{3\times 3} = \boldsymbol{A}\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B}$. One again we enforce the pseudo-measurement model $\boldsymbol{h}_{3\times 3} = \boldsymbol{0}$. A UKF with a state matrix instead of state vector can directly handle measurement models in matrix forms [11]. The pseudo-measurements can also be converted to a vector form as shown in [9] and then estimated using a UKF. Using dual quaternions we rewrite Eq. 23 in an alternate form, which would ultimately result in a linear pseudo-measurement, thus allowing us to use a linear Kalman filter for state estimation.

Let $\hat{\boldsymbol{a}}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{b}}$ be the dual quaternions corresponding to $\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{B}$ respectively. Eq. 23 can be rewritten as

$$\hat{\boldsymbol{a}} \otimes \hat{\boldsymbol{x}} - \hat{\boldsymbol{x}} \otimes \hat{\boldsymbol{b}} = \hat{\boldsymbol{0}}. \tag{24}$$

Using Eq. 8, Eq. 24 can be written as

$$\begin{aligned}\hat{\boldsymbol{0}} =&(\widetilde{\boldsymbol{a}}_r + \epsilon\widetilde{\boldsymbol{a}}_d) \otimes (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) - (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) \otimes (\widetilde{\boldsymbol{b}}_r + \epsilon\widetilde{\boldsymbol{b}}_d),\\ =& \left(\widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_r - \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_r\right) +\\ & \epsilon\left(\widetilde{\boldsymbol{a}}_d \odot \widetilde{\boldsymbol{q}}_r + \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_d - \widetilde{\boldsymbol{q}}_d \odot \widetilde{\boldsymbol{b}}_r - \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_d\right). \end{aligned} \tag{25}$$

Hence we have

$$\widetilde{\boldsymbol{0}} = \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_r - \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_r \tag{26}$$

$$\widetilde{\boldsymbol{0}} = \widetilde{\boldsymbol{a}}_d \odot \widetilde{\boldsymbol{q}}_r + \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_d - \widetilde{\boldsymbol{q}}_d \odot \widetilde{\boldsymbol{b}}_r - \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_d. \tag{27}$$

Eq. 26 has a form very similar to Eq. 17, with the only difference being that the scalar parts of $\widetilde{\boldsymbol{a}}_r, \widetilde{\boldsymbol{b}}_r$ are not 0. If $\widetilde{\boldsymbol{a}}_r = a_0 + \boldsymbol{a}_r$ and $\widetilde{\boldsymbol{b}}_r = b_0 + \boldsymbol{b}_r$, using Eq. 1 we rewrite Eq. 26 as

$$\boldsymbol{H}_r\widetilde{\boldsymbol{q}}_r = \boldsymbol{0}, \text{ where} \tag{28}$$

$$\boldsymbol{H}_r = \begin{bmatrix} a_0 - b_0 & -(\boldsymbol{a}_r - \boldsymbol{b}_r)^T \\ \boldsymbol{a}_r - \boldsymbol{b}_r & (\boldsymbol{a}_r + \boldsymbol{b}_r)^\times + (a_0 - b_0)\,\boldsymbol{I}_3 \end{bmatrix}_{4\times 4}. \tag{29}$$

The pseudo-measurement is

$$\boldsymbol{h}_r = \boldsymbol{H}_r\widetilde{\boldsymbol{q}}_r, \tag{30}$$

and the pseudo-measurement model is $\boldsymbol{h}_r = \boldsymbol{0}$. The uncertainty associated with $\boldsymbol{h}_r$ is derived in section IV-C3.

Similar to section IV-A, we use the estimated value of $\widetilde{\boldsymbol{q}}_r$ to estimate $\boldsymbol{t}$. Using Eq. 9, Eq. 27 can be rewritten as

$$\widetilde{\boldsymbol{0}} = \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r - \widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_r + \widetilde{\boldsymbol{\sigma}}_1, \tag{31}$$

where $\widetilde{\boldsymbol{\sigma}}_1 = 2\widetilde{\boldsymbol{a}}_d \odot \widetilde{\boldsymbol{q}}_r - 2\widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_d$. Multiplying both sides of Eq. 31 with $\widetilde{\boldsymbol{q}}_r^*$, we obtain:

$$\begin{aligned}\widetilde{\boldsymbol{0}} &= \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_t - \widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_r \odot \widetilde{\boldsymbol{q}}_r^* + \widetilde{\boldsymbol{\sigma}}_1 \odot \widetilde{\boldsymbol{q}}_r^*,\\ &= \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_t - \widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{\sigma}}_2 + \widetilde{\boldsymbol{\sigma}}_3, \end{aligned} \tag{32}$$

where $\widetilde{\boldsymbol{\sigma}}_2 = \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_r \odot \widetilde{\boldsymbol{q}}_r^*$ and $\widetilde{\boldsymbol{\sigma}}_3 = \widetilde{\boldsymbol{\sigma}}_1 \odot \widetilde{\boldsymbol{q}}_r^*$. The structure of Eq. 32 is similar to Eq. 26, with the only differences being the addition of $\widetilde{\boldsymbol{\sigma}}_3$ term and the scalar part of $\widetilde{\boldsymbol{q}}_t$ is 0. If $\widetilde{\boldsymbol{\sigma}}_2 = \sigma_2^0 + \boldsymbol{\sigma}_2$, using Eq. 1 we rewrite Eq. 32 as

$$\boldsymbol{0} = \boldsymbol{H}_t\boldsymbol{t} + \widetilde{\boldsymbol{\sigma}}_3, \text{ where} \tag{33}$$

$$\boldsymbol{H}_t = \begin{bmatrix} -(\boldsymbol{a}_r - \boldsymbol{\sigma}_2)^T \\ (\boldsymbol{a}_r + \boldsymbol{\sigma}_2)^\times + \left(a_0 - \sigma_2^0\right)\boldsymbol{I}_3 \end{bmatrix}_{4\times 3}. \tag{34}$$

Unlike the case discussed in section IV-A, $\boldsymbol{t}$ cannot always be directly obtained from the estimated $\widetilde{\boldsymbol{q}}_r$. This is because estimation of $\boldsymbol{t}$ would require inversion of $\boldsymbol{H}_t$, which need not always be invertible as it is formed from arbitrary sensor measurements. As shown in section V, a linear Kalman filter is employed with the following pseudo-measurement model to estimate $\boldsymbol{t}$,

$$\boldsymbol{h}_t = \boldsymbol{H}_t\boldsymbol{t} + \widetilde{\boldsymbol{\sigma}}_3. \tag{35}$$

We enforce the pseudo-measurement model $\boldsymbol{h}_t = \boldsymbol{0}$. The uncertainty associated with $\boldsymbol{h}_t$ is derived in section IV-C3.

C. Uncertainty in pseudo-measurements

In order to estimate the uncertainties associated with the pseudo-measurements as well as the translational vector described in the previous sections, we make use of an important result from stochastic theory [14, pp. 90–91], [5, Appendix A] described in Proposition 1.

Proposition 1. *Let us consider $\boldsymbol{b} \in \mathbb{R}^m$ and $\boldsymbol{c} \in \mathbb{R}^n$ which are sequences with zero mean. Let $\boldsymbol{h} \in \mathbb{R}^n$, $\boldsymbol{x} \in \mathbb{R}^l$ and a linear matrix function $\boldsymbol{G}(\cdot) : \mathbb{R}^l \to \mathbb{R}^{n\times m}$, such that $\boldsymbol{h} = \boldsymbol{G}(\boldsymbol{x})\boldsymbol{b}+\boldsymbol{c}$. Assume that $\boldsymbol{x}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are independent. Then $\boldsymbol{\Sigma}^{\boldsymbol{h}}$ is given by*

$$\boldsymbol{\Sigma}^{\boldsymbol{h}} = \boldsymbol{G}(\boldsymbol{x})\boldsymbol{\Sigma}^{\boldsymbol{b}}\boldsymbol{G}^T(\boldsymbol{x}) + \boldsymbol{N}(\boldsymbol{\Sigma}^{\boldsymbol{b}} \circledast \boldsymbol{\Sigma}^{\boldsymbol{x}})\boldsymbol{N}^T + \boldsymbol{\Sigma}^{\boldsymbol{c}}, \tag{36}$$

where $\circledast$ is the Kronecker product, $\boldsymbol{\Sigma}^{\{\cdot\}}$ is the uncertainty associated with $\{\cdot\}$ and $\boldsymbol{N} \in \mathbb{R}^{n\times lm}$ is defined as follows

$$\boldsymbol{N} \triangleq [\boldsymbol{G}_1 \ \boldsymbol{G}_2 \ \cdots \ \boldsymbol{G}_m].$$

$\boldsymbol{G}_i \in \mathbb{R}^{n\times m}$ is obtained from the following identity:

$$\boldsymbol{G}_i\boldsymbol{x} = \boldsymbol{G}(\boldsymbol{x})\boldsymbol{e}_i,$$

where $\boldsymbol{e}_i$ is the unit vector in $\mathbb{R}^m$ with 1 at position i and 0 everywhere else.

1) Uncertainty in pseudo-measurement for estimating the rotation in Case I: To find the uncertainty in the linear pseudo-measurement, we rewrite $\boldsymbol{h}$ from Eq. 20 in the following form

$$\begin{aligned}\boldsymbol{h} &= \boldsymbol{H}(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{q}_1, \boldsymbol{q}_2)\widetilde{\boldsymbol{q}}_r,\\ &= \boldsymbol{G}(\widetilde{\boldsymbol{q}}_r)\boldsymbol{v}_{\text{true}}, \quad \text{where } \boldsymbol{v}_{\text{true}} = (\boldsymbol{p}_1^T, \boldsymbol{p}_2^T, \boldsymbol{q}_1^T, \boldsymbol{q}_2^T)^T\\ &= \begin{bmatrix}\boldsymbol{G}_1 & -\boldsymbol{G}_1 & \boldsymbol{G}_2 & -\boldsymbol{G}_2\end{bmatrix}\boldsymbol{v}_{\text{true}}. \end{aligned} \tag{37}$$

In Eq. 37, $\boldsymbol{G}_1 = \begin{bmatrix} -\boldsymbol{q}_r^T \\ -\boldsymbol{q}_r^\times + q^0\boldsymbol{I}_3 \end{bmatrix}$ and $\boldsymbol{G}_2 = \begin{bmatrix} \boldsymbol{q}_r^T \\ -\boldsymbol{q}_r^\times - q^0\boldsymbol{I}_3 \end{bmatrix}$, where $\widetilde{\boldsymbol{q}}_r = q^0 + \boldsymbol{q}_r$ are obtained from Eq. 1. Eq. 37 is the pseudo-measurement for a noise-free sensor measurement $\boldsymbol{v}_{\text{true}}$. If $\boldsymbol{v}$ is the sensor measurement with noise $\delta\boldsymbol{v}$, then

$$\delta\boldsymbol{v} \triangleq \boldsymbol{v} - \boldsymbol{v}_{\text{true}}. \tag{38}$$

Solving for $\boldsymbol{v}_{\text{true}}$ from Eq. 38 and substituting in Eq. 37 yields:

$$\begin{aligned}\boldsymbol{h}(\widetilde{\boldsymbol{q}}_r) &= \boldsymbol{G}(\boldsymbol{v} - \delta\boldsymbol{v})\\ &= \boldsymbol{G}\boldsymbol{v} + \boldsymbol{\nu}_1, \end{aligned} \tag{39}$$

where $\boldsymbol{\nu}_1 = -\boldsymbol{G}\left(\widetilde{\boldsymbol{q}}_r\right)\delta\boldsymbol{v}$ is a zero mean noise. From Eq. 39, the uncertainty in the pseudo measurement $\boldsymbol{\Sigma}^{\boldsymbol{h}}$ can be obtained using Eq. 36.

2) Uncertainty in translation for Case I: The expression for $\boldsymbol{t}$ assuming perfect measurements $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ is given in Eq. 22. In the presence of noise in the measurements, similar to the derivation of Eq. 38, we obtain from Eq. 21

$$\boldsymbol{t} = \frac{\boldsymbol{p}_1 + \boldsymbol{p}_2}{2} - \text{vec}\left(\widetilde{\boldsymbol{q}}_r \odot \frac{\widetilde{\boldsymbol{q}}_1 + \widetilde{\boldsymbol{q}}_2}{2} \odot \widetilde{\boldsymbol{q}}_r^*\right) + \boldsymbol{\nu_2}, \quad \text{where,}$$

$$\boldsymbol{\nu_2} = -\frac{\delta\boldsymbol{p}_1 + \delta\boldsymbol{p}_2}{2} + \text{vec}\left(\widetilde{\boldsymbol{q}}_r \odot \frac{\delta\widetilde{\boldsymbol{q}}_1 + \delta\widetilde{\boldsymbol{q}}_2}{2} \odot \widetilde{\boldsymbol{q}}_r^*\right). \tag{40}$$

From Eq. 40, $\boldsymbol{\nu_2}$ is a zero mean noise with variance $\boldsymbol{\Sigma}^{\boldsymbol{t}} \in \mathbb{R}^{3\times3}$,

$$\boldsymbol{\Sigma}^{\boldsymbol{t}} = \frac{\boldsymbol{\Sigma}^{\boldsymbol{p}_1} + \boldsymbol{\Sigma}^{\boldsymbol{p}_2}}{4} + \boldsymbol{\Sigma}^{\boldsymbol{\tau}}, \tag{41}$$

where $\boldsymbol{\tau} = \text{vec}\left(\widetilde{\boldsymbol{q}}_r \odot \frac{\delta\widetilde{\boldsymbol{q}}_1 + \delta\widetilde{\boldsymbol{q}}_2}{2} \odot \widetilde{\boldsymbol{q}}_r^*\right)$, $\delta\widetilde{\boldsymbol{q}}_i = 0 + \delta\boldsymbol{q}_i$. $\boldsymbol{\Sigma}^{\boldsymbol{\tau}}$ is computed using Eq. 36 as shown below

$$\begin{aligned}\boldsymbol{\tau} =& \text{vec}\left(\widetilde{\boldsymbol{q}}_r \odot \frac{\delta\widetilde{\boldsymbol{q}}_1 + \delta\widetilde{\boldsymbol{q}}_2}{2} \odot \widetilde{\boldsymbol{q}}_r^*\right) \\ =& \text{vec}\left(\widetilde{\boldsymbol{q}}_r \odot (0 + \boldsymbol{\sigma})\right) = -\boldsymbol{G}_2\boldsymbol{\sigma}\end{aligned}$$

where $\tilde{\boldsymbol{\sigma}} = \frac{1}{2}\begin{bmatrix}\boldsymbol{G}_3 & \boldsymbol{G}_3\end{bmatrix}\begin{bmatrix}\delta\widetilde{\boldsymbol{q}}_1 \\ \delta\widetilde{\boldsymbol{q}}_2\end{bmatrix}$ and $\boldsymbol{G}_3 = \begin{bmatrix}\boldsymbol{q}_r^\times + q^0\boldsymbol{I}_3\end{bmatrix}$ is obtained from Eq. 1. Eq. 36 is then used to find $\boldsymbol{\Sigma}^{\boldsymbol{\sigma}}, \boldsymbol{\Sigma}^{\boldsymbol{\tau}}$.

3) Uncertainty in pseudo-measurement models for Case II: For pose based measurements, there are two pseudo-measurements corresponding to estimation to $\widetilde{\boldsymbol{q}}_R$ and $\boldsymbol{t}$. Eq. 30 and Eq. 35 are rewritten in the following form

$$\boldsymbol{h}_r = \boldsymbol{G}_r\boldsymbol{u}_{\text{true}}, \tag{42}$$

$$\boldsymbol{h}_t = \boldsymbol{G}_t\boldsymbol{w}_{\text{true}} + \boldsymbol{\sigma}_3, \tag{43}$$

where $\boldsymbol{G}_r = \begin{bmatrix} q_0 & -\boldsymbol{q}_r^T & -q_0 & \boldsymbol{q}_r^T \\ \boldsymbol{q}_r & q_0\boldsymbol{I}_3 - \boldsymbol{q}_r^\times & \boldsymbol{q}_r & -q_0\boldsymbol{I}_3 - \boldsymbol{q}_r^\times \end{bmatrix}$, $\boldsymbol{u}_{\text{true}} = (a_0, \boldsymbol{a}_r^T, b_0, \boldsymbol{b}_r^T)^T$, $\boldsymbol{G}_t = \begin{bmatrix} 0 & -\boldsymbol{t}^T & 0 & \boldsymbol{t}^T \\ \boldsymbol{t} & -\boldsymbol{t}^\times & \boldsymbol{t} & -\boldsymbol{t}^\times \end{bmatrix}$ and $\boldsymbol{w}_{\text{true}} = (a_0, \boldsymbol{a}_r^T, \sigma_2^0, \boldsymbol{\sigma}_2^T)^T$.

Eq. 42 and Eq. 43 are the pseudo-measurements for noise free sensor measurements $\boldsymbol{u}_{\text{true}}$, $\boldsymbol{w}_{\text{true}}$. If $\boldsymbol{u}$ and $\boldsymbol{w}$ are the sensor measurements with noise $\delta\boldsymbol{u}$ and $\delta\boldsymbol{w}$ respectively, then $\boldsymbol{h}_r = \boldsymbol{G}_r\boldsymbol{u} + \boldsymbol{\nu}_3$, $\boldsymbol{h}_t = \boldsymbol{G}_t\boldsymbol{w} + \boldsymbol{\sigma}_3 + \boldsymbol{\nu}_4$, where $\boldsymbol{\nu}_3 = -\boldsymbol{G}_r\delta\boldsymbol{u}$ and $\boldsymbol{\nu}_4 = -\boldsymbol{G}_t\delta\boldsymbol{w} - \delta\boldsymbol{\sigma}_3$ are zero mean noise with covariance $\boldsymbol{\Sigma}^{\boldsymbol{h}_r}$ and $\boldsymbol{\Sigma}^{\boldsymbol{h}_t}$ respectively, which can be obtained using Eq. 36.

V. Kalman filter equations

As shown in Eq. 17 and in Eq. 25, $\widetilde{\boldsymbol{q}}_r$ and $\boldsymbol{t}$ can be estimated in a decoupled manner. In this work, we formulate a Kalman filter that first estimates $\widetilde{\boldsymbol{q}}_r$ and then uses the estimated $\widetilde{\boldsymbol{q}}_r$ to estimate $\boldsymbol{t}$. For Case I, $\boldsymbol{t}$ and $\boldsymbol{\Sigma}^{\boldsymbol{t}}$ can be directly estimated from Eq. 22 and Eq. 41 upon estimating $\widetilde{\boldsymbol{q}}_r$ and $\boldsymbol{\Sigma}^{\widetilde{\boldsymbol{q}}_r}$. However for Case II, a Kalman filter is used to estimate the mean and uncertainty in $\boldsymbol{t}$.

The state vector of the Kalman filter that is used to estimate $\widetilde{\boldsymbol{q}}_r$ is $\boldsymbol{x}_k = \widetilde{\boldsymbol{q}}_r$, $\boldsymbol{x}_k \in \mathbb{R}^4$.. The state vector is initialized with a suitable guess for mean and uncertainty. In the absence of a good initial guess, the state is initialized to $\boldsymbol{x}_0 = (1,0,0,0)^T$ with a large initial uncertainty. If the uncertainty in rotation is known in terms of some other parametrizations such as Euler angles, then the uncertainty is propagated to the space of quaternions using a Jacobian mapping as shown in [22].

Since the $SE(3)$ element to be estimated is time-invariant, the process model is static, *i.e.*, $\boldsymbol{x}_{k|k-1} = \boldsymbol{x}_{k-1|k-1}$. Upon obtaining $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ $(i = 1, 2)$ for Case I, or $\hat{\boldsymbol{a}}, \hat{\boldsymbol{b}}$ for Case II, we formulate the pseudo-measurements $\boldsymbol{h}(\boldsymbol{x}_{k|k-1}) = \boldsymbol{H}_k\boldsymbol{x}_{k|k-1}$. The observation matrix $\boldsymbol{H}_k$ is given by Eq. 19 for Case I and by Eq. 29 for Case II. The measurement uncertainty $\boldsymbol{\Sigma}_k^{\boldsymbol{h}}$ is then calculated as shown in section IV-C1 for Case I and section IV-C3 for Case II.

The state is updated using standard equations of the Kalman filter [15]

$$\boldsymbol{x}_{k|k} = \boldsymbol{x}_{k|k-1} - \boldsymbol{K}_k\left(\boldsymbol{H}_k\boldsymbol{x}_{k|k-1}\right), \tag{44}$$

$$\boldsymbol{\Sigma}_{k|k}^{\boldsymbol{x}} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{\Sigma}_{k|k-1}^{\boldsymbol{x}}, \tag{45}$$

where $\boldsymbol{K}_k = \boldsymbol{\Sigma}_{k|k-1}^{\boldsymbol{x}}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{\Sigma}_{k|k-1}^{\boldsymbol{x}}\boldsymbol{H}^T + \boldsymbol{\Sigma}_k^{\boldsymbol{h}}\right)^{-1}$.

It has already been discussed that $\widetilde{\boldsymbol{q}}_r$ is a unit quaternion; which implies that the state vector has to be a unit vector. This requirement is not enforced by the equations of the Kalman filter directly. However, there are three methods to enforce unit-normalization of state vector (1) including the constraint as an additional pseudo-measurement [2], (2) reducing the dimension of the state vector by substituting $q_0 = \sqrt{1 - q_1^2 - q_2^2 - q_3^2}$ [27], (3) normalizing the state vector at the end of each update step [5]. The first two methods result in non-linear measurement models, which defeats our purpose of developing equations for a truly linear filter. As a result we resort to the third method of normalizing the state vector after every update and suitably scaling the uncertainty,

$$\boldsymbol{x}_{k|k}^* = \frac{\boldsymbol{x}_{k|k}}{\left\|\boldsymbol{x}_{k|k}\right\|}, \quad \Sigma_{k|k}^{\boldsymbol{x}^*} = \frac{\Sigma_{k|k}^{\boldsymbol{x}}}{\left\|\boldsymbol{x}_{k|k}\right\|^2}. \tag{46}$$

Such an approach has been shown to estimate efficiently in [20] and [10].

Upon estimating $\boldsymbol{x}_{k|k}$ and $\Sigma_{k|k}^{\boldsymbol{x}}$, Eq. 22 and Eq. 41 are used to estimate $\boldsymbol{t}_{k|k}$ and $\Sigma_{k|k}^{\boldsymbol{t}}$, for Case I. For Case II, we initiate another Kalman filter whose process model is static as in the case of $\widetilde{\boldsymbol{q}}_r$. The measurement model is also linear as in the case of $\widetilde{\boldsymbol{q}}_r$. The equation for the measurement model is as shown in Eq. 35. The observation matrix is evaluated at the estimated value of $\widetilde{\boldsymbol{q}}_r$.

VI. Results

We apply the filtering method developed in the earlier sections to two examples: rigid registration and sensor calibration representing Case I and Case II respectively. Simulation as well as experimental results are provided in the following sections.

A. Rigid registration

The rigid-registration problem can be defined as finding the $SE(3)$ element that aligns points in one reference frame to the points in another reference frame. Usually points in one frame are computed from a CAD model of the object and points in the other frame are estimated from images, position sensors, laser range scanners, etc.

Iterative closest point (ICP) is one of the most popular methods to perform rigid-registration [3]. A number of variants to the ICP have been introduced [25, 30]. ICP and most of its variants are batch processing tools; i.e., one needs to wait for all the measurements to be collected before estimating the transformation. Also in the presence of noisy data, ICP and most of its variants have been observed to perform poorly [24]. As a result, online estimation techniques have been developed to account for noise in the measurements [24, 21, 11].

In this work, we apply DQF method to register 100 points randomly sampled from the surface of a "Stanford bunny", to its CAD model. We first assume that the point correspondence is known and estimate the registration with DQF, whose actual registration parameters are all 0s. We sample 1000 initial registration estimates uniformly drawn from large initial errors in position, for $x, y, z \in [-10000, 10000]$mm and orientation $\theta_x, \theta_y, \theta_z \in [-180, 180]$deg. From Fig. 1, we observe that DQF correctly estimates the registration for all the initial estimates. Following this, we perform two more experiments with noise added to the sampled points. The noise is uniformly sampled from $[-2, 2]$mm along each axis, in one case and $[-3, 3]$mm in the other. Fig. 1 shows that the RMS error for all the estimates is only due to the noise in the measurements and its magnitude matches well with the noise added to the points. Thus, when the point correspondences are known, DQF accurately estimates the registration parameters even in presence of very high errors in the initial estimate.

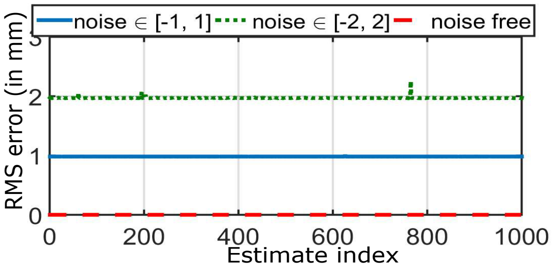

Fig. 1. RMS error upon estimating registration parameters with DQF for 1000 runs with different initial estimates, when the point correspondence is known. Three experiments were carried out: 1) noise uniformly sampled from [-1,1], 2) noise uniformly sampled from [-2,2] and 3) no noise was added. DQF accurately estimates he registration parameters in all cases

Following this observation, we now perform registration in a more realistic scenario where point correspondence is unknown. Point correspondence to the CAD model is found using a closest point rule as in [3, 24, 21]. We repeat the exercise by adding noise to the points and then estimating the transformation. In both the cases, we compare the dual quaternion based filtering to an EKF based estimator [24] and a UKF based estimator [21]. We choose an initial guess of zero rotation and zero translation and an initial covariance of $\boldsymbol{\Sigma}_0^{\tilde{\boldsymbol{q}}_r} = 5\boldsymbol{I}_4$ for rotation and $\boldsymbol{\Sigma}_0^{\boldsymbol{t}} = 100\boldsymbol{I}_3$ for translation. DQF, EKF and UKF are implemented with 40 initial starts obtained by locally perturbing the initial guess by translation sampled uniformly from $[-15\ 15]$mm along each axis and a rotation sampled uniformly from $[-30\ 30]$deg along each axis. Since the problem has several local minima, using multiple initial guesses improves the chances of finding the global minimum.

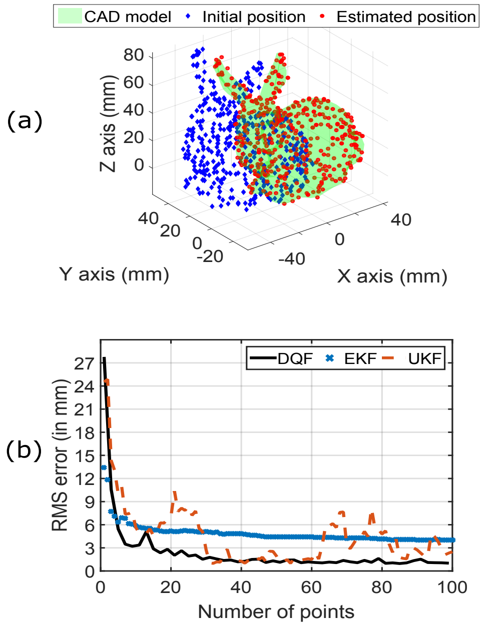

Fig. 2. (a) Initial position and DQF estimated position of 100 points are shown against the CAD model of the "Stanford bunny". DQF accurately registers the points to the CAD model. (b) A plot of the RMS error wrt number of points for DQF, EKF and UKF. DQF and EKF converge quickly, while UKF takes a while to converge. DQF however converges to lower RMS error, with computation time an order of magnitude lower than EKF and UKF as shown in Table I.

Fig. 2(a) shows the CAD model of the Stanford bunny in green. The blue diamond markers show the initial guess for the location of the points and the red circular markers show the DQF estimated location of the points. Fig. 2(b) shows the RMS error versus number of points used to estimate the parameters. The RMS error decreases with the usage of more point measurements. DQF and EKF converge to a smaller RMS error at about 10 points, while the UKF takes many more points to converge. First four rows of Table I show the actual registration parameters and the estimated registration parameters. The right column of Table I shows the time taken by the filters to update for 100 point measurements. DQF converges an order of magnitude faster than EKF and UKF and also has the lowest RMS error.

Fig. 3 shows the results for the case where the sampled points are corrupted with a noise uniformly sampled from $[-2\ 2]$mm along each axis. DQF accurately registers the points to the CAD model as shown in the last three rows of Table I. DQF once again performs better than EKF and UKF, and takes lower computational time [1].

[1] The computational time taken is calculated for a code running on MATLAB R2015a software from MathWorks, running on a ThinkPad T450s computer with 8 GB RAM.

TABLE I
REGISTRATION RESULTS FOR STANFORD BUNNY

No noise	x	y	z	θ_x	θ_y	θ_z	RMS	Time
	(mm)	(mm)	(mm)	(deg)	(deg)	(deg)	(mm)	(s)
Actual	22	-23	20	15	-10	-10	–	–
DQF	22.54	-21.52	20.03	17.28	-9.94	-10.15	1.12	23.44
EKF	22.35	-26.39	21.11	11.43	-11.44	-14.76	3.88	155.02
UKF	21.36	-23.89	18.94	16.39	-5.95	-10.55	2.47	247.56
With noise								
DQF	22.34	-24.22	18.79	13.37	-9.09	-10.18	2.70	47.05
EKF	20.29	-26.09	20.69	8.76	-12.79	-8.08	3.81	324.23
UKF	20.08	-24.78	14.6	11.90	-6.08	-8.04	4.80	510.73

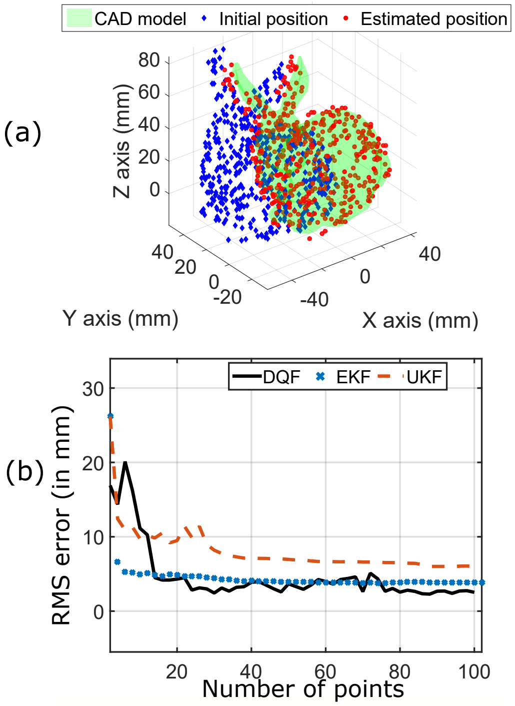

Fig. 3. (a) Initial position and estimated position of 100 points with added noise are shown against the CAD model of the "Stanford bunny". DQF estimates the registration parameters accurately even in the presence of noise. (b) A plot of the RMS error wrt number of points for DQF, EKF and UKF. DQF and EKF converge quickly, but UKF takes a while to converge. Overall, all the three filters converge closely to one another, with the DQF performing marginally better. The DQF converges with computation time an order of magnitude lower than the other two as shown in Table I.

B. Sensor calibration

The sensor calibration problem is as follows: given the pose of two bodies $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$, defined with respect to two different inertial frames: $\{1\}$ and $\{2\}$, we would like to estimate the rigid transformation between the two bodies, by tracking $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$, where the index i denotes an instance of time. Fig. 4(a) shows the various frames described above. This problem can be described as $\boldsymbol{A}_{ij}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}_{ij}$, where $\boldsymbol{A}_{ij} = \boldsymbol{A}_i^{-1}\boldsymbol{A}_j$ and $\boldsymbol{B}_{ij} = \boldsymbol{B}_i^{-1}\boldsymbol{B}_j$. $\boldsymbol{X}$ is the rigid transformation between the two bodies which needs to be estimated as shown in Fig. 4(b).

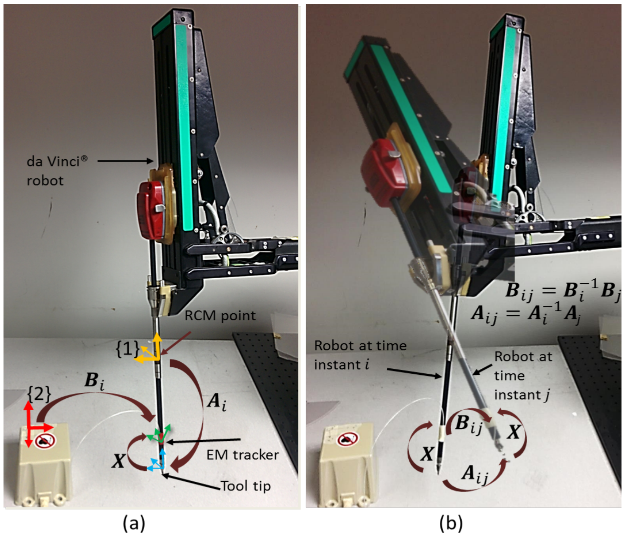

Fig. 4. (a) The setup shows a da Vinci robot with an EM tracker rigidly attached to the tool. The reference frame for the EM tracker is shown in red. The reference frame for the robot is located at its remote center of motion (RCM), shown in yellow. The pose of the tip of the robot, $\boldsymbol{A}_i$ is shown in blue and the pose of the sensor, $\boldsymbol{B}_i$ is shown in green. $\boldsymbol{X}$ is the transformation between the tip of the robot and the EM tracker. (b) The robot is shown at two time instances i and j. $\boldsymbol{A}_{ij}$ is obtained from kinematics and $\boldsymbol{B}_{ij}$ is obtained from the EM tracker measurements. The unknown to be solved for is $\boldsymbol{X}$, which can be posed in the form: $\boldsymbol{A}_{ij}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}_{ij}$.

If the measurements are noise-free, then $\boldsymbol{X}$ can be obtained analytically from a pair of measurements: $\boldsymbol{A}_{12}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}_{12}$ and $\boldsymbol{A}_{23}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}_{23}$ [23, 4]. But sensors are seldom noise-free, and hence several optimization based approaches exist to solve this problem [13, 26], whose solution drives many applications [7, 1, 8, 12]. Recently Faion et al [9] developed a filtering based solution to this problem, which could perform online estimation using a UKF to estimate the pose which is parameterized using axis-angle and Cartesian parameters. We compare our DQF to this UKF based estimation. We also develop an EKF based estimation using the measurement model described in [9] for a second comparison.

C. Simulation

We first tested our algorithm with simulated data and then with data collected from real experiments. For the simulated case, we first generate 500 random poses for the tool tip, $\boldsymbol{A}_i$ $(i = 1, \ldots, 500)$. We then choose a ground truth $SE(3)$ element $\boldsymbol{X}$ to generate the corresponding poses for the EM tracker $\boldsymbol{B}_i$. We initialize the filters to zero translation and zero rotation with an initial covariance of $\boldsymbol{\Sigma}_0^{\widetilde{\boldsymbol{q}}_r} = 5\boldsymbol{I}_4$ for rotation and $\boldsymbol{\Sigma}_0^{\boldsymbol{t}} = 100\boldsymbol{I}_3$ for translation. We assume that the correspondence between the sensed poses is known. Such an assumption is reasonable as the sensor measurements can be easily time-synchronized. If this synchronization is not possible, correlation between the sensor measurements can be obtained as shown in [19].

The $SE(3)$ elements estimated by DQF, EKF and UKF are shown in Table II along with computation time for each algorithm and the error in position and orientation parameters.

DQF provides faster estimates and is more accurate, especially in the translation estimation, compared to UKF and EKF.

Following this, we perturb $\boldsymbol{B}_i$ that is computed from the ground truth $\boldsymbol{X}$. The pose is perturbed by a translation uniformly sampled from the interval $[-2\ 2]$mm along each axis and a rotation uniformly sampled from $[-10\ 10]$deg along each axis. The estimated parameters are shown in Table II. Once again DQF estimated faster and is more accurate than UKF and EKF.

D. Experimental validation

In order to test our formulation with real data, we use an experimental setup as shown in Fig. 4(a), which consists of a da Vinci® surgical robot (Intuitive Surgical Inc., Mountain View, CA) and an electromagnetic (EM) tracking sensor (trakSTAR™ from Ascension Technologies, Burlington,VT). The tracker is rigidly attached to a known point on the tool of the robot. The robot is then telemanipulated and the position and orientation of the tip of the robot is measured from the kinematics. The position and orientation of the EM tracker with respect to the inertial frame attached to the magnetic field generator is also simultaneously recorded. We then use DQF, EKF and UKF to estimate the transformation between the frames of the tip of the robot and the frame of the EM tracker. Last three rows of Table II shows the parameters as estimated by DQF, EKF and UKF.

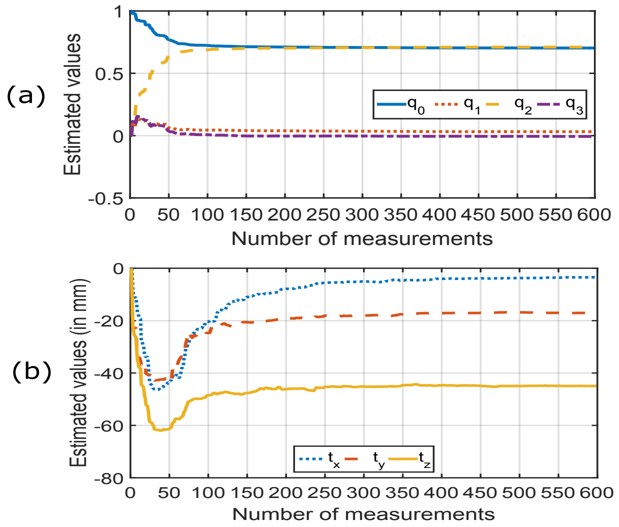

Fig. 5. (a) The plot shows the estimated value of the quaternion that represents the rotation. The values converge at around 100 measurements. (b) The plot shows the estimated value of the translation vector. The values converge at around 200 measurements.

Fig. 5 shows the values of the quaternion and the translation vector as estimated by the dual quaternion filter. The estimated values converge at around 100 measurements for rotation and 200 measurements for translation. Since the rotation estimation does not depend upon translation estimation, we can stop running the filter that estimates rotation after convergence and continue to run the translation filter until convergence. We observe that it takes around 0.08s for the DQF estimate to converge which is roughly 5 times faster than EKF and 15 times faster than UKF, while being more accurate than both EKF and UKF. Since the estimation is close to real time, we implement this algorithm in an online manner to estimate $SE(3)$ elements as needed.

TABLE II
SENSOR CALIBRATION RESULTS

Simulation: No noise in sensor measurements							
	x	y	z	θ_x	θ_y	θ_z	Time
	(mm)	(mm)	(mm)	(deg)	(deg)	(deg)	(s)
Actual	5.73	8.59	11.46	10.00	-16.00	35.00	–
DQF	5.73	8.59	11.46	9.96	-15.95	34.91	0.25
EKF	3.38	1.82	5.25	10.09	-15.93	35.05	1.20
UKF	3.56	10.59	10.81	9.98	-15.98	35.05	3.11
Simulation: With noise in sensor measurements							
DQF	5.59	8.22	11.38	9.95	-15.95	34.81	0.24
EKF	-3.48	4.22	8.36	10.14	-16.01	35.01	1.15
UKF	5.89	10.44	10.01	10.83	-16.81	34.81	3.20
Robot experiments							
Actual	-4	-20	45	105	88	109	–
DQF	-4.10	-17.50	-45.10	105.55	87.88	108.69	0.27
EKF	-3.60	-22.00	-45.10	105.97	86.15	107.08	1.39
UKF	-4.40	-14.1	-47.50	132.11	87.05	135.01	3.70

VII. CONCLUSION

In this work, we have developed linear measurement models to be used with Kalman filters for the estimation of $SE(3)$ elements. This was possible due to our choice of using dual quaternions to represent $SE(3)$ elements and combining multiple sensor measurements simultaneously. All the information contained in the non-linear update model was encoded in the linear measurement model and its corresponding uncertainty, which happens to be state dependent in this case. Since the dependence on the state was found to be linear, results from stochastic theory were used to determine the exact expressions for the uncertainty. We show that the new linear measurement model allows for decoupled estimation of rotation and translation using independent Kalman filters. The decoupled estimation potentially has the advantage of running in-parallel and accelerating the estimation process.

We have shown through simulations and experiments that the dual quaternion-based linear filtering is capable of estimating the $SE(3)$ more accurately with less computation time compared to state-of-the-art filtering methods for $SE(3)$ estimation. These characteristics of the dual quaternion-based filter, make it an ideal candidate to be used in applications that require real-time estimation of $SE(3)$ elements such as sensor calibration, localization and manipulation. Adapting the method for applications that involve estimation of time varying $SE(3)$ elements, whose process model may not be linear, will be a subject of future work.

ACKNOWLEDGMENTS

This work has been funded through the National Robotics Initiative by NSF grant IIS-1426655.

References

[1] Martin Kendal Ackerman, Andrew Cheng, Bernard Shiffman, Emad Boctor, and Gregory Chirikjian. Sensor calibration with unknown correspondence: Solving AX= XB using Euclidean-group invariants. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1308–1313. IEEE, 2013.

[2] IY Bar-Itzhack, J Deutschmann, and FL Markley. Quaternion normalization in additive EKF for spacecraft attitude determination. page 403, 1991.

[3] Paul J Besl and Neil D McKay. Method for registration of 3-D shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[4] Homer H Chen. A screw motion approach to uniqueness analysis of head-eye geometry. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 145–151. IEEE, 1991.

[5] D. Choukroun, I.Y. Bar-Itzhack, and Y. Oshman. Novel quaternion Kalman filter. *IEEE Transactions on Aerospace and Electronic Systems*, 42(1):174–190, Jan 2006. ISSN 0018-9251. doi: 10.1109/TAES.2006.1603413.

[6] Clifford. Preliminary Sketch of Biquaternions. *Proceedings of the London Mathematical Society*, s1-4(1):381–395, 1871. doi: 10.1112/plms/s1-4.1.381.

[7] Yuchao Dai, Jochen Trumpf, Hongdong Li, Nick Barnes, and Richard Hartley. Rotation averaging with application to camera-rig calibration. In *Proceedings of the 9th Asian conference on Computer Vision-Volume Part II*, pages 335–346. 2009.

[8] Fadi Dornaika and Radu Horaud. Simultaneous robot-world and hand-eye calibration. *IEEE Transactions on Robotics and Automation*, 14(4):617–622, 1998.

[9] F. Faion, P. Ruoff, A. Zea, and U.D. Hanebeck. Recursive Bayesian calibration of depth sensors with non-overlapping views. In *15th International Conference on Information Fusion (FUSION)*, pages 757–762, July 2012.

[10] James Samuel Goddard and Mongi A Abidi. Pose and motion estimation using dual quaternion-based extended Kalman filtering. In *Photonics West'98 Electronic Imaging*, pages 189–200. International Society for Optics and Photonics, 1998.

[11] Søren Hauberg, François Lauze, and Kim Steenstrup Pedersen. Unscented Kalman filtering on Riemannian manifolds. *Journal of Mathematical Imaging and Vision*, 46(1):103–120, 2013.

[12] Radu Horaud and Fadi Dornaika. Hand-eye calibration. *The International Journal of Robotics Research (IJRR)*, 14(3):195–210, 1995.

[13] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, 1987.

[14] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[15] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[16] Ben Kenwright. *A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3D character hierarchies*. Václav Skala-UNION Agency, 2012.

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

[18] Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter. *Nonlinear Kalman filtering for force-controlled robot tasks*. Springer-Verlag Berlin, 2004.

[19] Elmar Mair, Michael Fleps, Michael Suppa, and Darius Burschka. Spatio-temporal initialization for IMU to camera registration. In *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 557–564. IEEE, 2011.

[20] João Luís Marins, Xiaoping Yun, Eric R Bachmann, Robert B McGhee, and Michael J Zyda. An extended Kalman filter for quaternion-based orientation estimation using MARG sensors. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 4, pages 2003–2011. IEEE, 2001.

[21] Mehdi Hedjazi Moghari and Purang Abolmaesumi. Point-based rigid-body registration using an unscented Kalman filter. *IEEE Transactions on Medical Imaging*, 26(12):1708–1728, 2007.

[22] K.E. Nicewarner and A.C. Sanderson. A general representation for orientational uncertainty using random unit quaternions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1161–1168 vol.2, May 1994.

[23] Frank C Park and Bryan J Martin. Robot sensor calibration: solving AX= XB on the Euclidean group. *IEEE Transactions on Robotics and Automation*, 10(5), 1994.

[24] Xavier Pennec and Jean-Philippe Thirion. A framework for uncertainty and validation of 3-D registration methods based on points and frames. *International Journal of Computer Vision*, 25(3):203–229, 1997.

[25] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152. IEEE, 2001.

[26] Yiu Cheung Shiu and Shaheen Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX= XB. *IEEE Transactions on Robotics and Automation*, 5(1):16–29, 1989.

[27] Malcolm D Shuster. The quaternion in Kalman filtering. *Advances in the Astronautical Sciences*, 85:25–37, 1993.

[28] Davide Spinello and Daniel J Stilwell. Nonlinear estimation with state-dependent Gaussian observation noise. *IEEE Transactions on Automatic Control*, 55(6):1358–1366, 2010.

[29] Rangaprasad Arun Srivatsan, Matthew Travers, and Howie Choset. Using Lie algebra for shape estimation of medical snake robots. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 3483–3488. IEEE, 2014.

[30] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, 2013.

[31] Shiuh-Ku Weng, Chung-Ming Kuo, and Shu-Kang Tu. Video object tracking using adaptive Kalman filter. *Journal of Visual Communication and Image Representation*, 17(6):1190–1208, 2006.

[32] Simon Winkelbach, Sven Molkenstruck, and Friedrich M Wahl. Low-cost laser range scanner and fast surface registration approach. In *Pattern Recognition*, pages 718–728. Springer, 2006.