Distributed Dual Quaternion Extended Kalman Filtering for Spacecraft Pose Estimation

Mathias Hudoba de Badyn [*], Jonas Binz [†], Andrea Iannelli [‡], and Roy S. Smith [§]

Department of Technology Systems, University of Oslo, 2027, Kjeller, Norway
Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, Swiss Federal Instutute of Technology (ETH), Zürich, 8092, Switzerland
Institute for Systems Theory and Automatic Control, University of Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart

In this paper, a distributed dual-quaternion multiplicative extended Kalman filter for the estimation of poses and velocities of individual satellites in a fleet of spacecraft is analyzed. The proposed algorithm uses both absolute and relative pose measurements between neighbouring satellites in a network, allowing each individual satellite to estimate its own pose and that of its neighbours. By utilizing the distributed Kalman consensus filter, a novel sensor and state-estimate fusion procedure is proposed that allows each satellite to improve its own state estimate by sharing data with its neighbours over a communication link. A leader-follower approach, whereby only a subset of the satellites have access to an absolute pose measurement is also examined. In this case, followers rely solely on the information provided by their neighbours, as well as relative pose measurements to those neighbours. The algorithm is tested extensively via numerical simulations, and it is shown that the approach provides a substantial improvement in performance over the scenario in which the satellites do not cooperate. A case study of satellites swarming an asteroid is presented, and the performance in the leader-follower scenario is also analyzed.

Nomenclature

$\mathbf{I}_{n \times k}$	=	identity matrix in $\mathbb{R}^{n \times k}$
$\mathbf{0}_{n \times k}$	=	zero matrix in $\mathbb{R}^{n \times k}$
$\mathbf{1}_n$	=	vector of all ones in $\mathbb{R}^n$
I	=	inertial frame
B, B_i	=	body frame, body frame of satellite i
A_i	=	body frame of the IMU of satellite i
$\mathbb{H}, \mathbb{H}^u$	=	set of quaternions/unit quaternions $\{q \in \mathbb{H} \mid qq^* = 1\}$
$A(q)$	=	rotation matrix corresponding to quaternion q
$q_s, \bar{q}$	=	scalar and vector parts of quaternion q
$q_{B/I}$	=	unit quaternion defining the attitude transformation I to B
$r^I_{B/I}$	=	vector quaternion defining the position of B in respect to I expressed in I
$\overline{r^I_{B/I}}$	=	three-dimensional vector defining the position of B in respect to I expressed in I
ϵ	=	a dual number
$\mathbb{H}_d, \mathbb{H}_d^u$	=	set of dual quaternions, and unit dual quaternions $\{\mathbf{q} \in \mathbb{H}_d \mid \mathbf{q}\mathbf{q}^* = 1\}$
$\mathbf{q}, q_r, q_d$	=	dual quaternion, real/dual parts of dual quaternion
$\mathbf{q}_{B/I}, \overline{\mathbf{q}_{B/I}}$	=	unit, and reduced unit dual quaternion defining the pose transformation from I to B
$\omega^B_{B/I}, \omega^B_{B/I}$	=	vector and reduced dual velocity of B compared to I expressed in B
$\omega^B_{B/I}$	=	angular velocity of B compared to I expressed in B [rad/s]

---

[*]Associate Professor, Department of Technology Systems, University of Oslo, Gunnar Randers vei 19, 2027, Kjeller, Norway. `mathias.hudoba@its.uio.no`, AIAA Member

[†]Master's Student, Automatic Control Laboratory, Physikstrasse 3, 8092, Zürich, Switzerland.

[‡]Assistant Professor, Institute for Systems Theory and Automatic Control, University of Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart, Germany `andrea.iannelli@ist.uni-stuttgart.de`

[§]Professor, Automatic Control Laboratory, Physikstrasse 3, 8092, Zürich, Switzerland. `rsmith@control.ee.ethz.ch`, AIAA Associate Fellow

$v_{B/I}^{B}$	=	linear velocity of B compared to I expressed in B [m/s]
$[\|q\|]_R, [\|q\|]_L$	=	function from $\mathbb{R}^4 \to \mathbb{R}^{4\times4}$ denoting the right/left quaternion multiplication
$[\|\mathbf{q}\|]_R, [\|\mathbf{q}\|]_L$	=	function from $\mathbb{R}^8 \to \mathbb{R}^{8\times8}$ denoting the right/left dual quaternion multiplication
x_i	=	the state of satellite i
Δx_i	=	Kalman state update of satellite i
P_i	=	the estimation covariance matrix of satellite i
F_i	=	state transition matrix of satellite i
G_i	=	process noise matrix of satellite i
z_i	=	measurement vector of satellite i
$z_{i,k}$	=	measurement vector of satellite i expressed in the states of satellite k
H_i	=	observation matrix of satellite i
$H_{i,k}$	=	observation matrix of satellite i expressed in the states of satellite k
$\hat{\mathbf{q}}_{B/I}, \mathbf{q}_{B/I,m}$	=	estimate/measurement of the pose transformation from I to B
$\hat{\omega}_{B/I}^{B}, \omega_{B/I,m}^{B}$	=	estimate/measurement of the dual velocity of B compared to I expressed in B
$\mathbf{b}_\omega, \hat{\mathbf{b}}_\omega$	=	dual bias/estimate of dual bias
$\mathbf{J}_{8\times8}$	=	extended inertia matrix
$\mathcal{G}$	=	communication graph of the satellite network
V	=	indexed set of nodes of $\mathcal{G}$ corresponding to individual satellites
L, F	=	sets of leader/follower nodes
E	=	set of edges of $\mathcal{G}$ corresponding to commmunication links between satellites
N_i	=	set of all neighbours of satellite i, excluding i
J_i	=	set of all neighbours of satellite i, including i
$\Lambda_{i,k}$	=	set of all neighbours which satellite i and satellite k have in common
$\overrightarrow{\hat{q}_i}$	=	estimate of all satellite poses $\in J_i$ done by satellite i
$\overrightarrow{\hat{b}_i}$	=	estimate of all dual bias' $\in J_i$ done by satellite i
$\overrightarrow{\hat{\omega}_i}$	=	estimate of all dual velocities $\in J_i$ done by satellite i
$\overrightarrow{\hat{\omega}_{i,m}}$	=	dual velocities measurements $\in J_i$ done by satellite i
y_i	=	aggregated measurement vector of satellite i
S_i	=	aggregated covariance matrix of satellite i
$\mu_{q,i}$	=	weight of the soft consensus of satellite i on the quaternion
$\mu_{r,i}$	=	weight of the soft consensus of satellite i on the position
$\mu_{b,i}$	=	weight of the soft consensus of satellite i on the bias

I. Introduction

Terrestrial constraints such as light pollution, the day-night cycle, and an increasingly more crowded low-Earth orbit all point towards the future of astronomy being on spaceborne satellites in deep space [1]. Distributing the task of observation over multiple satellites has the potential to increase observational capacity by extending the baseline for imaging [2], or providing a disturbance-free environment for novel telescope types like laser interferometers [3, 4]. Such satellite systems require the ability for every satellite in the fleet to point at the same object cooperatively, or maintain some relative formation, including both position and attitude (in other words, *pose*) constraints. This requires the development of novel distributed pose control and estimation algorithms.

Attitude control for spacecraft using quaternions is a mature field [5–8]. Conversely, the paradigm of dual quaternions, which allows a compact representation of attitude and position (or the pose) of a spacecraft, is a new area of research. Novel control scenarios using dual quaternions include distributed control methods [9], powered-descent guidance [10], data-driven techniques [11], as well as static distributed estimation [12]. Dual quaternions also provide a natural formalism for multi-spacecraft proximity operations such as rendezvous [13–15] and multi-spacecraft formation flight [16].

Much effort has been dedicated to spacecraft estimation [7, 17]. Kalman filtering based on the unit quaternion is well-understood, and has been used in many different applications, including NASA spacecraft [6, 7, 18]. The success of the quaternion multiplicative extended Kalman filter (Q-MEKF) can be attributed to different key components of the filter. In contrast to Euler angles, unit quaternions provide a non-singular representations of attitude, while using

the minimum number of parameters*. Additionally, updating attitudes in the unit quaternion framework has a lower computational complexity in comparison to Euler angles.

Dual quaternion multiplicative extended Kalman filtering (DQ-MEKF), on the other hand, is a more recent development and has not been used to the same degree as the Q-MEKF [8, 19]. However, the compact representation of the pose via the dual quaternion make their use in spacecraft pose estimation appealing for several reasons. Firstly, the kinematic and dynamic equations in the dual quaternion setting have the same form as that in the attitude-only quaternion setting, meaning that much of intuition on quaternion-based estimation and control can be re-interpreted in the dual quaternion algebra. Secondly, visual navigation systems, such as Position Sensing Diodes [20, 21], or camera-based relative measurements [19, 22, 23], inherently couple attitude and translation. Therefore, a unified framework for pose estimation is useful when considering these types of sensors.

Distributed Kalman filtering has also been an area of rapid recent development [24]. A distributed estimation procedure based on dual quaternions was proposed in [12], wherein each sensor node estimates the pose through the Kalman filter and then updates its estimate by solving a minimization problem which is represented by a function of the estimates and covariances of its neighbours. A DQ-MEKF was used in a two-satellite setting for pose estimation and control in [25]. A more general approach is that of the Kalman-consensus filter, where a consensus term is added after the update step of the Kalman filter [26–28]. This approach also assumes a multi-sensor model in which all sensor nodes estimate the same state — other works with similar assumptions include [26, 29, 30]. A multi-robot distributed Kalman filter where each robot estimates only its own state (in 2D), but uses relative measurements between neighbouring robots, is proposed by [31]. A clear gap in the literature is a distributed version multiplicative extended Kalman filter in the dual quaternion framework. We address this here, for general network topologies, and for several sensor models.

The contributions of this paper are as follows. First, we present a novel distributed dual quaternion extended Kalman filter operating on a fleet of satellites defined on a network described by a graph. Each satellite has access to a pose measurement, but can also measure the pose of its neighbours relative to its own pose. Facilitated by communication between neighbouring satellites, measurement, covariance and estimation data can be exchanged, by which the pose estimates can be further improved by sensor and covariance fusion. In particular, we propose two sensor and state estimate fusion steps, which we dub 'soft' and 'hard' consensus, which allow neighbouring satellites to share information about their state estimates and relative measurements and improve the overall performance of each satellite in the fleet. Briefly, soft consensus is a distributed averaging of the estimates with neighbouring satellites, and hard consensus includes a measurement and covariance fusion step. The proposed distributed algorithm is then tested in numerical simulations and compared to the setting in which each satellite computes its own state estimate without any cooperation. The simulations show that the proposed algorithm results in smaller Root Mean Square (RMS) errors in pose and velocity compared to a satellite fleet without any cooperation.

Second, we examine the filter algorithm in the case where there is a set of 'leaders', which have access to a 'global' pose measurement with respect to an inertial frame (e.g., a star-tracker for attitude and a laser range-finder/GPS to a nearby body for position), and a set of followers which have only access to relative measurements with respect to their neighbours. We derive the necessary algebraic steps for filtering, sensor and measurement fusion, and communication. Numerical experiments show that we are able to remove the absolute pose measurement from a substantial part of the satellite fleet without degradation in performance of the estimator. Third, we provide an open-source repository [32] containing the code to reproduce the numerical examples in the paper, which can also be used for more general distributed filtering problems.

The paper is organized as follows. We introduce preliminaries in §II, and we derive our distributed filter, including the hard and soft consensus terms, and the leader-follower setting in §III. Numerical experiments are presented in §IV, and the paper is concluded in §V. Lengthy calculations are delegated to appendices.

II. Preliminaries

Our notation follows that of [19]. Additionally, for matrices $A_1, \ldots, A_n$ we define $\mathrm{blkdiag}\{A_1, \ldots, A_n\}$ to be the matrix formed by placing $A_1, \ldots, A_n$ on the block-diagonal. A summary of the DQ-MEKF from [19] is provided in Appendix B of supplementary material [33].

---

*It should be noted that the Q-MEKF, which only uses the vector part of the quaternion, can still be singular if the error between the attitude and its estimate is greater than 180°

A. Information Form Extended Kalman Filter

Consider the nonlinear dynamical system,

$$\dot{x}(t) = f(x(t), t, u(t)) + g(x(t), t)w(t) \tag{1}$$

with state variable $x(t) \in \mathbb{R}^n$, input $u(t) \in \mathbb{R}^q$, and Gaussian process noise $w(t) \in \mathbb{R}^p$ with $E[w(t)] = \mathbf{0}_p$ and $\mathrm{Cov}[w(t)] = Q(t)$. The function $f : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^q \mapsto \mathbb{R}^n$ maps the state and input to its derivative, and $g : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^{n \times p}$ maps the noise onto the dynamics.

Consider an estimate of $x(t)$ defined as $\hat{x}(t) := E[x(t)]$, and the covariance of that estimate $P(t) := \mathrm{Cov}[x(t)]$. The estimate and its covariance satisfy respectively,

$$\dot{\hat{x}}(t) = E[f(x(t), t, u(t))] \tag{2}$$

$$\dot{P}(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t), \tag{3}$$

where

$$F(t) = \left. \frac{\partial f(x, t, u(t))}{\partial x} \right|_{\hat{x}(t)} \tag{4a}$$

$$G(t) = g(\hat{x}(t), t). \tag{4b}$$

Eq. (2) is usually approximated as,

$$\dot{\hat{x}}(t) \approx f(\hat{x}(t), t, u(t)). \tag{5}$$

Consider the measurement $z(t_k) \in \mathbb{R}^m$ given by the model,

$$z(t_k) = h(x(t)) + v(t), \tag{6}$$

where $v(t)$ is measurement noise represented as a Gaussian process with $E[v(t)] = \mathbf{0}_m$ and $\mathrm{Cov}[v(t)] = R(t)$.

We can write the discrete measurement update as,

$$\hat{x}^+(t_k) = \hat{x}^-(t_k) + \Delta\hat{x}(t_k) \tag{7}$$

where $\hat{x}^-$ is the predicted value given by the time update from the previous timestep, $\hat{x}^+$ is the predicted value after the measurement update, and $\Delta\hat{x}(t_k)$ is the Kalman state update given by,

$$\Delta\hat{x}(t_k) = K(t_k)[z(t_k) - \hat{z}(t_k)], \tag{8}$$

where the predicted measurement is,

$$\hat{z}(t_k) = h(\hat{x}(t)), \tag{9}$$

and the Kalman gain $K \in \mathbb{R}^{m \times n}$ is given by

$$K(t_k) = P^-(t_k)H^T(t_k) \left[H(t_k)P^-(t_k)H^T(t_k) + R \right]^{-1}, \tag{10}$$

where $H(t_k)$ is the Jacobian of the measurement function,

$$H(t_k) = \left. \frac{\partial h(x)}{\partial x} \right|_{\hat{x}^-(t_k)}. \tag{11}$$

After the measurement, the covariance is updated by

$$P^+(t_k) = (\mathbf{I} - K(t_k)H(t_k))P^-(t_k). \tag{12}$$

Finally, $\hat{x}^+(t_k)$ is propagated forward to t_{k+1} via the dynamics (5), yielding $\hat{x}^-(t_{k+1})$.

Following [34], we introduce the information form Kalman filter, modified to the current setting of an EKF. The information form Kalman filter has some benefits in comparison to the usual Kalman filter in the distributed case;

4

in particular it allows batch measurement processing with lower complexity by avoiding several superfluous matrix inversions. From Eq. (1) to Eq. (7) the steps of the filter are the same. We introduce:

$$u(t_k) = H^T(t_k)R^{-1}z(t_k) \tag{13}$$

$$U(t_k) = H^T(t_k)R^{-1}H(t_k) \tag{14}$$

$$M(t_k) = \left(P^{-1}(t_k) + U(t_k)\right)^{-1}. \tag{15}$$

Finally, the Kalman state update is computed by

$$\Delta\hat{x}(t_k) = M(t_k)\left(u(t_k) - U(t_k)\hat{x}(t_k)\right). \tag{16}$$

The matrix M can be seen as the *a posteriori* covariance.

B. Dual Quaternions

For reasons of brevity, we assume that the reader is familiar with the basic properties and operations of quaternions, as in [17]. These operations have algebraic representations that are useful for computation; in this setting, the quaternion is represented as a vector $q \in \mathbb{R}^4$. In order to multiply two quaternions a and b, we define the following

$$[a] = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix}^T := \begin{bmatrix} a_0 \\ \bar{a} \end{bmatrix}, \quad [a]^\times := \begin{bmatrix} 0 & \mathbf{0}_{1\times3} \\ \mathbf{0}_{3\times1} & \bar{a}^\times \end{bmatrix}, \quad \bar{a}^\times := \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \tag{17}$$

where we have adopted the scalar-first convention. We can respectively write the *left and right quaternion multiplication operators* as,

$$[|a|]_L := \begin{bmatrix} a_0 & -\bar{a}^T \\ \bar{a} & [\tilde{a}]_L \end{bmatrix}, \quad [|a|]_R := \begin{bmatrix} a_0 & -\bar{a}^T \\ \bar{a} & [\tilde{a}]_R \end{bmatrix} \tag{18}$$

where $[\tilde{a}]_L = a_0\mathbf{I}_{3\times3} + \bar{a}^\times$ and $[\tilde{a}]_R = a_0\mathbf{I}_{3\times3} - \bar{a}^\times$. Then, quaternion multiplication and the cross product are given by $[ab] = [|a|]_L[b] = [|b|]_R[a] \in \mathbb{R}^4$ and $[a \times b] = [a]^\times[b] \in \mathbb{R}^4$, respectively.

The unit quaternions are used for attitude estimation as they describes a rotation from one coordinate frame (for example, the inertial frame I) to another coordinate frame (for example, the body frame B). Here, q^* denotes the quaternion conjugate. Formally, a rotation amplitude θ and a rotation axis $\overrightarrow{n} \in \mathbb{R}^3$, which describes the rotation from frame I to B, can be described as the quaternion $q_{B/I}$

$$q_{B/I} = [\cos(\theta/2), \overrightarrow{n}^T \sin(\theta/2)]^T. \tag{19}$$

The inverse rotation is then simply $q^*_{B/I}$. The scalar part of the quaternion can be computed as,

$$q_0 = \sqrt{1 - \|\bar{q}\|^2}. \tag{20}$$

Dual quaternions are an extension of quaternions that encode both the position and attitude of a rigid body. The *dual quaternion* is defined by,

$$\mathbf{q} = q_r + \epsilon q_d \qquad q_r, q_d \in \mathbb{H}, \tag{21}$$

where the *dual number* ϵ satisfies $\epsilon^2 = 0$ and $\epsilon \neq 0$. The basic operations on dual quaternions are analogous to that of the quaternions, and are as follows:
Addition:

$$\mathbf{a} + \mathbf{b} = (a_r + b_r) + \epsilon(a_d + b_d) \in \mathbb{H}_d \tag{22}$$

Multiplication by scalar:

$$\lambda\mathbf{a} = (\lambda a_r) + \epsilon(\lambda a_d) \in \mathbb{H}_d \tag{23}$$

5

Multiplication:

$$\mathbf{ab} = (a_r b_r) + \epsilon(a_r b_d + a_d b_r) \in \mathbb{H}_d \tag{24}$$

Conjugation:

$$\mathbf{a}^* = a_r^* + \epsilon a_d^* \in \mathbb{H}_d \tag{25}$$

Cross Product:

$$\mathbf{a} \times \mathbf{b} = a_r \times b_r + \epsilon(a_d \times b_r + a_r \times b_d) \in \mathbb{H}_d. \tag{26}$$

As in the case of the quaternion, it is convenient to represent the dual quaternion operations in algebraic terms. Define the following vectors and operators:

$$[\mathbf{q}] = \begin{bmatrix} [q_r] \\ [q_d] \end{bmatrix} \in \mathbb{R}^8 \qquad [|a|]_R = \begin{bmatrix} [|a_r|]_R & \mathbf{0}_{4\times4} \\ [|a_d|]_R & [|a_r|]_R \end{bmatrix} \in \mathbb{R}^{8\times8}$$

$$[|a|]_L = \begin{bmatrix} [|a_r|]_L & \mathbf{0}_{4\times4} \\ [|a_d|]_L & [|a_r|]_L \end{bmatrix} \in \mathbb{R}^{8\times8} \qquad \bar{\mathbf{a}} = \begin{bmatrix} \bar{a}_r \\ \bar{a}_d \end{bmatrix} \in \mathbb{R}^6 \tag{27}$$

$$[\tilde{\mathbf{a}}] = \begin{bmatrix} [\tilde{a}_r] & \mathbf{0}_{3\times3} \\ [\tilde{a}_d] & [\tilde{a}_r] \end{bmatrix} \in \mathbb{R}^{6\times6} \qquad \bar{\mathbf{a}}^\times = \begin{bmatrix} \bar{a}_r^\times & \mathbf{0}_{3\times3} \\ \bar{a}_d^\times & \bar{a}_r^\times \end{bmatrix} \in \mathbb{R}^{6\times6}.$$

Then, we can represent dual quaternion multiplication as $[\mathbf{ab}] = [|a|]_L[\mathbf{b}] = [|b|]_R[\mathbf{a}] \in \mathbb{R}^8$.

The set of unit dual quaternions is of particular interest in the setting of *pose* estimation and control, where the pose is a representation of both position and attitude.

Consider an object whose B frame is at $\overline{r_{B/I}^I}$ with respect to the I frame, expressed in the I frame. Its rotation from the origin to its body frame is described by an angle θ and its rotation axis, $\overrightarrow{n}$. The (real) rotation quaternion q_r is computed using Eq. (19). Then, the unit dual quaternion describing the pose of the object is

$$q_d = \frac{1}{2} r_{B/I}^I q_r, \quad r_{B/I}^I = (0, \overline{r_{B/I}^I}), \quad \mathbf{q} = q_r + \epsilon q_d \in \mathbb{H}_d^u \tag{28}$$

C. Rigid Body Model in Dual Quaternions

We now introduce the kinematics and dynamics for a single rigid body using dual quaternions [16, 19].

1. Dual Kinematics

The unit dual quaternion which defines the transformation from inertial frame I to the body frame B of the rigid body is

$$\mathbf{q}_{B/I} = q_r + \epsilon q_d = q_r + \epsilon\frac{1}{2} r_{B/I}^I q_r = q_r + \epsilon\frac{1}{2} q_r r_{B/I}^B. \tag{29}$$

The dual velocity $\omega_{B/I}^B \in \mathbb{H}_d^v$, which is a vector quaternion, is defined as

$$\omega_{B/I}^B = \begin{bmatrix} 0 \\ \omega_{B/I}^B \end{bmatrix} + \epsilon \begin{bmatrix} 0 \\ v_{B/I}^B \end{bmatrix} \tag{30}$$

where $\omega_{X/Y}^Z$ describes the *dual velocity* in the X frame compared to the Y frame, expressed in the Z frame. The real part describes the angular velocity, whereas the dual part describes the linear velocity. Using the dual velocity, the dual kinematics are derived using Eq. (21) as,
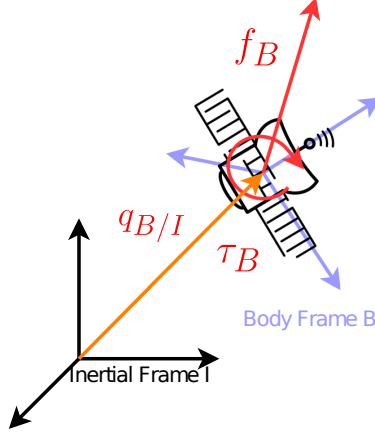
$$\dot{\mathbf{q}}_{B/I} = \dot{q}_r + \epsilon\dot{q}_d \tag{31}$$

$$= \frac{1}{2}\mathbf{q}_{B/I}\omega_{B/I}^B, \tag{32}$$

where the form of $\dot{q}_d$ is derived as,

$$\dot{q}_d = \frac{1}{2}v_{B/I}^I q_r + \frac{1}{2}r_{B/I}^I \dot{q}_r \tag{33}$$

$$= \frac{1}{2}q_r v_{B/I}^B + \frac{1}{2}q_d \omega_{B/I}^B. \tag{34}$$

Fig. 1   The dual quaternion $\mathbf{q}_{B/I}$ defines the pose of the body frame B with respect to the intertial frame I. Forces f_B and torques τ_B are applied in the body frame.

2. Dual Dynamics

All forces and torques from actuators are exerted on the satellite in the body frame; thus the dual force $\mathbf{f}_B$ is introduced as,

$$\mathbf{f}_B = \begin{bmatrix} 0 \\ \tau_B \end{bmatrix} + \epsilon \begin{bmatrix} 0 \\ f_B \end{bmatrix}. \tag{35}$$

An illustration is shown in Fig. 1. The extended inertia matrix is defined as

$$\mathbf{J}_{8\times8} = \text{blkdiag}\left\{1, m\mathbf{I}_{3\times3}, 1, \mathbf{J}\right\}, \tag{36}$$

where m is the mass of the object and $\mathbf{J}$ is its inertia matrix. Following [16], the dual dynamics are given by,

$$\omega_{B/I}^B = \mathbf{J}^{-1}(\mathbf{f}_B - \omega_{B/I}^B \times \mathbf{J}\omega_{B/I}^B). \tag{37}$$

D. Graph Theory and Distributed Kalman Filtering

When performing Kalman filtering across a network of distributed computational nodes with access to only local information, a centralized algorithm for state estimation is infeasible. Furthermore, each node may only have access to a measurement of a subset of the states of the system. These were some of the motivating factors for introducing distributed Kalman filtering for sensor networks in [27]. This is the foundation for the distributed approach of our present work and is therefore explained in detail here.

A graph describes the connectivity of a set of objects. In particular, a graph $G = (V, E)$ is a set of *nodes* $V = \{1, \ldots, l\}$ and a set of *edges* $E \subseteq V \times V$, where for $i, j \in V$, $ij \in E$ if and only if there is a connection between i and j. In the context of this paper, a 'connection' between satellites i and j denotes that satellite i and j can communicate, as well as measure relative poses of each other. For simplicity, we assume that G is *undirected*, in that if $ij \in E$, then $ji \in E$: if satellite i can communicate with j, then j can also communicate with i[†]. The *neighbourhood* N_i of $i \in V$ is defined as the set of $j \in V$ such that $ij \in E$. We later use a set $V_i := N_i \cup \{i\}$ to describe the neighbourhood of i if i is defined to be a neighbour of itself.

Let us assume a graph G with connectivity defined by an adjacency matrix $\mathcal{A}$, where each node i has a set of neighbours N_i. Furthermore, the nodes want to estimate the state x of the dynamical system given by Eq. (1)[‡].

Every node has an estimate $\hat{x}_i$ of the state x[§]. As such, the time update can be executed locally, using the EKF procedure described in §II.A. Every node propagates its estimate and covariance locally. However, in such a setting a

---

[†]If we were to consider directed graphs, then this would imply one-way communication and measurements between neighbouring satellites
[‡]In our work, each node will only estimate a subset of the states of the system. This will be elaborated on in §III.B.
[§]Later in the paper, we adapt the distributed Kalman filter so that each satellite i estimates its own state, and that of its neighbours

node may not be able to observe all states, or each node might have a different sensor. Even if all nodes had a full state measurement, it may be advantageous to fuse this information to get better estimates than one node might have by itself.

Let each node $i \in V$ have the sensor model

$$z_i(t_k) = h(x(t_k)) + v_i(t_k). \tag{38}$$

For 'local' (read: non-distributed) Kalman filtering, only this measurement would be used to compute the Kalman update. In the distributed setting [27], measurement, estimate and covariance data from the neighbours of i can be fused to improve the estimate $\hat{x}_i$ of i.

At each timestep t_k, each node i first does the time update of its estimate $\hat{x}_i$ and covariance M_i by using Eqns. (3), (4) and (5). Then, each node $i \in V$ takes a measurement z_i with the measurement model given in Eq. (38). Each node can now compute u_i and U_i with Eq. (13), which is then broadcast along with the state estimate to its neighbours. Explicitly, the data the neighbours receive is the packet,

$$\text{msg}_i = (u_i, U_i, \hat{x}_i^-). \tag{39}$$

Now the data and covariance matrices are aggregated in a data fusion step as,

$$y_i = \sum_{j \in N_i} u_j \qquad S_i = \sum_{j \in N_i} U_j \qquad M_i = \left(P_i^{-1} + S_i\right)^{-1} \tag{40}$$

and the Kalman state update is given by,

$$\hat{x}_i^+ = \hat{x}_i^- + M_i[y_i - S_i\hat{x}_i^-] + \mu_i \sum_{j \in N_i} \left(\hat{x}_j^- - \hat{x}_i^-\right) \tag{41}$$

where $\mu_i > 0$ is a tuning parameter. The assumptions surrounding the optimality of the state update in Eq. (41) and are discussed in [34].

III. Distributed Dual Quaternion Extended Kalman Filter

In this section, we present the distributed dual quaternion extended Kalman filter for spacecraft pose estimation. The derivation of the distributed filter is conducted by assuming that each satellite has access to absolute pose measurements, after which we elaborate on the *leader-follower* scenario where only some satellites have absolute pose measurements.

A. Problem Statement

Let us assume a fleet of l satellites, whose communication topology is described by a graph $\mathcal{G} = (V, E)$. Each node $i \in V$ represents a satellite, which has a set of neighbours N_i with $|N_i| = k_i$. We emphasize that V is an ordered set, i.e., each satellite has a numerical label $i \in V$, and we assume that the fleet knows its own label and those of its neighbours.

We assume that at each timestep, every satellite i takes k_i relative pose measurements in its own body frame, one for each neighbouring satellite. For now, we will also assume that each satellite takes one absolute pose measurement with respect to the inertial frame, and we consider cases where the satellites may or may not have access to its velocities and accelerations in the body frame. We will relax this assumption later in the paper.

The relative pose measurements can be taken by a visual sensor [20, 35–37]. The absolute pose measurements can be taken by a combination of star trackers and a Global Positioning System (GPS). The angular velocity and linear acceleration measurement can be taken via an IMU. Furthermore, we assume that each satellite has communication links to its neighbouring satellites for data exchange. Each satellite therefore only keeps track of its own state, and the state of its neighbours.

B. Derivation of the distributed DQ-MEKF (DDQ-MEKF)

In this setting, consider the set of satellites[¶] V with $|V| = l$. Each satellite $i \in V$ will only estimate its own pose and the pose of its neighbours, i.e., it can only estimate the poses of satellites $j \in V_i := N_i \cup i$. Furthermore, each node i shall compute the pose, the pose error, as well as the velocity measurement bias and velocity of the ith satellite. We can thus define the state and process noise vector of each satellite.

¶We will use 'nodes' and 'satellites' interchangeably from this point onwards
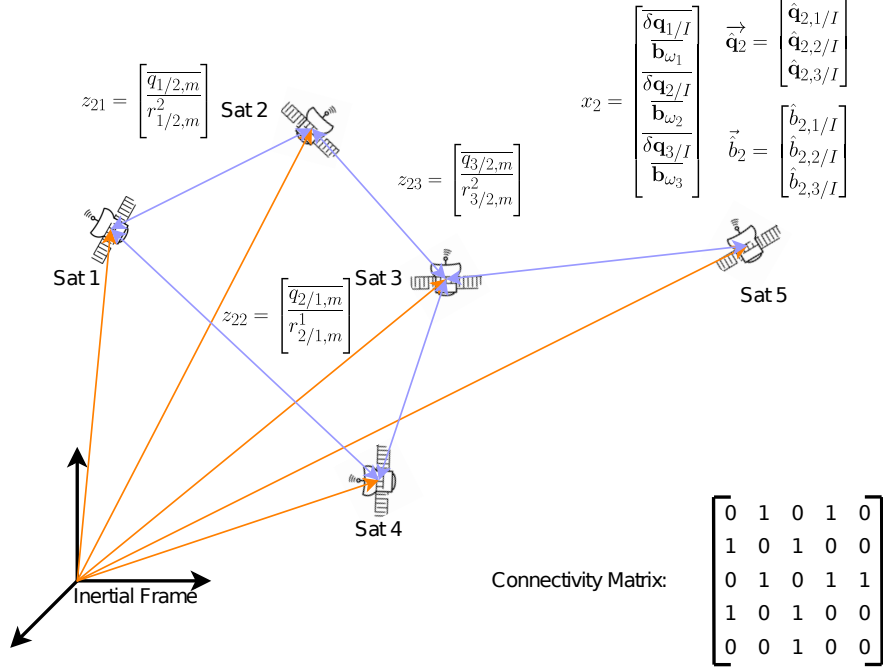
Fig. 2 Configuration of a fleet of $l = 5$ satellites. The states of satellite 2 are shown, where x_2 is the filter state, and $\overrightarrow{\hat{\mathbf{q}}_2}$ and $\overrightarrow{\hat{b}_2}$ are the respective estimated pose and bias of satellite 2 and its neighbours.

Let $i \in V$. We write $V_i = \{j_{i,1}, \ldots, j_{i,k_i+1}\} \subset V$, where $j_{i,k} \in V_i$ is the kth neighbour of i – note that due to the definition of V_i, satellite i is also treated as a neighbour of i. Since i knows the node labels of its neighbours, we can assume without loss of generality that V_i is sorted in ascending order of those labels. Then, we denote for each $i = 1, \ldots, l$ the state variables $x_i \in \mathbb{R}^{12(k_i+1)}$ and $w_i \in \mathbb{R}^{12(k_i+1)}$ given by,

$$
\begin{aligned}
x_i &= \begin{bmatrix} \overline{\delta \mathbf{q}}^T_{j_{i,1}/I} & \overline{\mathbf{b}}^T_{\omega\, j_{i,1}} & \cdots & \overline{\delta \mathbf{q}}^T_{j_{i,k_i+1}/I} & \overline{\mathbf{b}}^T_{\omega\, j_{i,k_i+1}} \end{bmatrix}^T \\
w_i &= \begin{bmatrix} \overline{\boldsymbol{\eta}}^T_{\omega\, j_{i,1}} & \overline{\boldsymbol{\eta}}^T_{\mathbf{b}_\omega\, j_{i,1}} & \cdots & \overline{\boldsymbol{\eta}}^T_{\omega\, j_{i,k_i+1}} & \overline{\boldsymbol{\eta}}^T_{\mathbf{b}_\omega\, j_{i,k_i+1}} \end{bmatrix}^T,
\end{aligned}
\tag{42}
$$

where we simplify the notation via $\overline{\delta \mathbf{q}}_{j_{i,k}/I} := \overline{\delta \mathbf{q}}_{B_{j_{i,k}}/I}$. Following the centralized filter described in Appendix B of the supplementary material [33], $\overline{\delta \mathbf{q}}_{j_{i,k}/I}$ is the error dual quaternion representing the pose error relative to a reference of the kth neighbour of i, $\overline{\mathbf{b}}_{\omega\, j_{i,k}}$ is the dual bias of the velocity, and w_i is a vector of process noises.

In contrast to the non-cooperative case in Appendix B of the supplementary material [33], the definition of the state and noise in Eqn (42) is a stacking of the states of satellite i and its neighbours. The top state is always the state of the satellite with the smallest nodal label. We further define the stacked pose estimate vector $\overrightarrow{\hat{\mathbf{q}}_i}$ and the stacked dual bias vector $\overrightarrow{\hat{b}_i}$ for $i = 1, \ldots, l$ as,

$$
\overrightarrow{\hat{\mathbf{q}}_i} = \begin{bmatrix} \hat{\mathbf{q}}^T_{i,j_{i,1}/I} & \cdots & \hat{\mathbf{q}}^T_{i,j_{i,k_i+1}/I} \end{bmatrix}^T \in \mathbb{R}^{8(k_i+1)}
\tag{43a}
$$

$$
\overrightarrow{\hat{b}_i} = \begin{bmatrix} \overline{\hat{\mathbf{b}}}^T_{\omega\, i,j_{i,1}} & \cdots & \overline{\hat{\mathbf{b}}}^T_{\omega\, i,j_{i,1}} \end{bmatrix}^T \in \mathbb{R}^{6(k_i+1)}.
\tag{43b}
$$

Again, to simplify notation we denote $\hat{\mathbf{q}}_{i,k/I} := \hat{\mathbf{q}}_{i,B_k/I}$ as the estimate that satellite i computes of the pose of neighbouring satellite k.

1. Time Update

At the beginning of each iteration, each satellite $i \in V$ takes a measurement $\omega^{B_i}_{B_i/I,m}$ (where the subscript m denotes 'measurement') of its own dual velocity in its body frame B_i, and propagates this information to its neighbours. To

reduce notational burden of notation, we write $\omega_{j_{i,k}/I,m} := \omega^{B_{j_{i,k}}}_{B_{j_{i,k}}/I,m}$ to denote the measured dual velocity of the kth neighbour of satellite i in the body frame of the kth neighbour of satellite i. Hence, each satellite $i \in V$ can construct the stacked dual velocity measurement as,

$$\overrightarrow{\omega}_{i,m} = \left[\overline{\omega_{j_{i,1}/I,m}}^T \quad \cdots \quad \overline{\omega_{j_{i,k_i+1}/I,m}}^T \right]^T \in \mathbb{R}^{6(k_i+1)}. \tag{44}$$

By removing the dual bias from the dual velocity, each $i \in V$ can compute the estimated stacked dual velocity via,

$$\overrightarrow{\hat{\omega}}_i = \overrightarrow{\omega}_{i,m} - \overrightarrow{\hat{b}}_i = \left[\overline{\hat{\omega}_{j_{i,1}/I,m}}^T \quad \cdots \quad \overline{\hat{\omega}_{j_{i,k_i+1}/I,m}}^T \right]^T \tag{45}$$

This yields the set of differential equations for the estimate,

$$\frac{d}{dt} \begin{bmatrix} \hat{\mathbf{q}}_{i,j_{i,1}/I} \\ \vdots \\ \hat{\mathbf{q}}_{i,j_{i,k_i+1}/I} \end{bmatrix} \approx \begin{bmatrix} \frac{1}{2}\hat{\mathbf{q}}_{i,j_{i,1}/I} \hat{\omega}_{j_{i,1}/I,m} \\ \vdots \\ \frac{1}{2}\hat{\mathbf{q}}_{i,j_{i,k_i+1}/I} \hat{\omega}_{j_{i,k_i+1}/I,m} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} [|\hat{\mathbf{q}}_{i,j_{i,1}/I}|]_L & & 0 \\ & \ddots & \\ 0 & & [|\hat{\mathbf{q}}_{i,j_{i,k_i+1}/I}|]_L \end{bmatrix} \begin{bmatrix} \hat{\omega}_{j_{i,1}/I,m} \\ \vdots \\ \hat{\omega}_{j_{i,k_i+1}/I,m} \end{bmatrix}, \tag{46}$$

from which the time update of the state is computed.

Next, the covariance of the state is propagated. Each node $i \in V$ must compute the Jacobians of the dynamics $F_{i,m}$ and $G_{i,m}$ for each of its neighbours $m \in V_i$ using Eq. (4). Explicitly, these are,

$$F_{i,m} = \begin{bmatrix} -\overline{\hat{\omega}_{i,j_{i,k}/I}}^\times & -\frac{1}{2}\mathbf{I}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \end{bmatrix} \text{ for } m = 1, \ldots k_i + 1, \tag{47a}$$

$$G_{i,m} = \begin{bmatrix} -\frac{1}{2}\mathbf{I}_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{I}_{6\times6} \end{bmatrix} \quad \text{for } m = 1, \ldots k_i + 1. \tag{47b}$$

The complete Jacobians of the dynamics and measurement models F_i and G_i of satellite $i \in V$ are computed by concatenating the aforementioned $F_{i,k}$ and $G_{i,k}$ matrices block-diagonally, yielding

$$F_i = \text{blkdiag}\{F_{i,m}\}_{m=1}^{k_i+1} \in \mathbb{R}^{12(k_i+1)\times12(k_i+1)} \tag{48a}$$

$$G_i = \text{blkdiag}\{G_{i,m}\}_{m=1}^{k_i+1} \in \mathbb{R}^{12(k_i+1)\times12(k_i+1)}. \tag{48b}$$

The stacked Q_i matrix therefore becomes,

$$Q_i = \text{blkdiag}\left\{ \tilde{Q}_{i,m} \right\}_{m=1}^{k_i+1} \tag{49}$$

$$\tilde{Q}_{i,m} = \text{blkdiag}\left\{ \overline{Q}_{\omega_{j_{i,m}}}, \overline{Q}_{\mathbf{b}_{\omega_{j_{i,m}}}} \right\}. \tag{50}$$

Next, each node $i \in V$ can propagage the covariance using Eq. (3) As stated above, the only communication needed between satellites is the exchange of the dual velocity measurement $\omega^{B_i}_{B_i/I,m}$. In the case where this information is not accessible (i.e., no gyro or linear acceleration measurement), no communication is necessary.

2. Measurement Update

We must fuse both the relative and absolute pose measurements in order to derive the measurement update step. As explained in §III.A, each satellite $i \in V$ has access to an absolute pose measurement $\mathbf{q}_{B_i/I,m}$, and k_i relative attitude and relative position measurements, $q_{B_k/B_i,m}$ and $r^{B_i}_{B_k/B_i,m}$, in its own body frame. We can thus denote the measurement vector z_i of satellite i as,

$$z_i = \left[\overline{q_{B_{n_{i,1}}/B_i,m}}^T \quad \overline{r^{B_i}_{B_{n_{i,1}}/B_i,m}}^T \quad \cdots \quad \overline{\hat{\mathbf{q}}^*_{i,i}\mathbf{q}_{B_i/I,m}}^T \quad \cdots \quad \overline{q_{B_{n_{i,k_i}}/B_i,m}}^T \quad \overline{r^{B_i}_{B_{n_{i,k_i}}/B_i,m}}^T \right]^T \tag{51}$$

$$:= \left[\overline{h_{q_{n_{i,1}/i}}(x_i)}^T \quad \overline{h_{r^i_{n_{i,1}/i}}(x_i)}^T \quad \cdots \quad \overline{h_{abs}(x_i)}^T \quad \cdots \quad \overline{h_{q_{n_{i,k_i}/i}}(x_i)}^T \quad \overline{h_{r^i_{n_{i,k_i}/i}}(x_i)}^T \right]^T + v_i. \tag{52}$$

10

Note that $z_i \in \mathbb{R}^{6(k_i+1)}$. In Equation (52), $h_{q_{n_{i,k}/i}}$ is the function which outputs the relative measurement from satellite i and its neighbour with index $n_{i,k}$ to the state x_i of satellite i. In a similar manner, $h_{r^i_{n_{i,k}/i}}$ gives the relative position. Only one entry is different, namely $\overline{\hat{\mathbf{q}}^*_{i,i}\mathbf{q}_{B_i/I,m}}$, which is the satellite's measurement of its own state. v_i is the measurement noise acting on all measurements.

The Jacobian H of the measurement function given in Eq. (52) can be viewed as the (linearized around zero error) mapping from the state to the measurement. In the case of Eq. (52), the Jacobian consists of three submatrices: the mapping of the absolute pose measurement to the state, the mapping from the relative attitude measurement to the state, and finally the mapping from the relative position measurement to the state. For sake of clarity, we derive the Jacobians of the individual measurement functions h in Appendix A.A, and proceed to the construction of the full Jacobian of the measurement in Eq. (52).

Full Jacobian of the Measurement:
We begin as follows. We define the following four matrices as,

$$\overline{H^i_{i,k}} := \begin{bmatrix} \frac{1}{2}\overline{H_{q_{i/k},\delta q_{i/I}}} & \mathbf{0}_{3\times3} \\ \frac{1}{4}\overline{H_{r^i_{k/i},\delta q_{i/I}}} & \frac{1}{4}\overline{H_{r^i_{k/i},\delta p_{i/I}}} \end{bmatrix}, \quad \overline{H^k_{i,k}} := \begin{bmatrix} \frac{1}{2}\overline{H_{q_{i/k},\delta q_{k/I}}} & \mathbf{0}_{3\times3} \\ \frac{1}{4}\overline{H_{r^i_{k/i},\delta q_{k/I}}} & \frac{1}{4}\overline{H_{r^i_{k/i},\delta p_{k/I}}} \end{bmatrix} \tag{53}$$

$$H^i_{i,k} := \begin{bmatrix} \frac{1}{2}H_{q_{i/k},\delta q_{i/I}} & \mathbf{0}_{4\times4} \\ \frac{1}{4}H_{r^i_{k/i},\delta q_{i/I}} & \frac{1}{4}H_{r^i_{k/i},\delta p_{i/I}} \end{bmatrix}, \quad H^k_{i,k} := \begin{bmatrix} \frac{1}{2}H_{q_{i/k},\delta q_{k/I}} & \mathbf{0}_{4\times4} \\ \frac{1}{4}H_{r^i_{k/i},\delta q_{k/I}} & \frac{1}{4}H_{r^i_{k/i},\delta p_{k/I}} \end{bmatrix}, \tag{54}$$

where the submatrices $H_a \in \mathbb{R}^{4\times4}, \overline{H_a} \in \mathbb{R}^{3\times3}$ corresponding to the Jacobians of the various measurement models of attitude and position are derived in Appendix A.A. The matrix $H^i_{i,k}$ can be understood as the part of the measurement Jacobian which transforms the state of satellite i to the relative measurement that satellite i takes of satellite k. On the other hand, $H^k_{i,k}$ is the part of the measurement Jacobian which transforms the state of satellite k, estimated from satellite i to the relative measurement that satellite i takes of satellite k. Here, we define $\overline{H_a} \in \mathbb{R}^{3\times3}$ as the matrix $H_a \in \mathbb{R}^{4\times4}$ with its first row and column (corresponding to the scalar part of the quaternion) deleted. Thus, $\overline{H_a}$ is congruent with the reduced state space defined in Eq. (42).

Finally, let $H_i := H_{i,i}$ and $\overline{H_i} := \overline{H_{i,i}}$. The stacked measurement Jacobian for each satellite i is given by,

$$H_i = \begin{bmatrix} H^{n_{i,1}}_{i,n_{i,1}} & \mathbf{0}_{8\times6} & \cdots & H^i_{i,n_{i,1}} & \mathbf{0}_{8\times6} & \cdots & \mathbf{0}_{8\times8} & \mathbf{0}_{8\times6} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{8\times8} & \mathbf{0}_{8\times6} & \cdots & \mathbf{I}_{8\times8} & \mathbf{0}_{8\times6} & \cdots & \mathbf{0}_{8\times8} & \mathbf{0}_{8\times6} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{8\times8} & \mathbf{0}_{8\times6} & \cdots & H^i_{i,n_{i,k_i}} & \mathbf{0}_{8\times6} & \cdots & H^{k_i}_{i,n_{i,k_i}} & \mathbf{0}_{8\times6} \end{bmatrix} \in \mathbb{R}^{8(k_i+1)\times14(k_i+1)} \tag{55}$$

$$\overline{H_i} = \begin{bmatrix} \overline{H^{n_{i,1}}_{i,n_{i,1}}} & \mathbf{0}_{6\times6} & \cdots & \overline{H^i_{i,n_{i,1}}} & \mathbf{0}_{6\times6} & \cdots & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \cdots & \mathbf{I}_{6\times6} & \mathbf{0}_{6\times6} & \cdots & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \cdots & \overline{H^i_{i,n_{i,k_i}}} & \mathbf{0}_{6\times6} & \cdots & \overline{H^{k_i}_{i,n_{i,k_i}}} & \mathbf{0}_{6\times6} \end{bmatrix} \in \mathbb{R}^{6(k_i+1)\times12(k_i+1)}, \tag{56}$$

where $\overline{H^{n_{i,j}}_{i,n_{i,j}}}$ is always in the j-th column and j-th row if $n_{i,j} < i$, and in the $j+1$-th column and $j+1$-th row otherwise. The matrix $\overline{H^i_{i,n_{i,j}}}$ is always in the i-th column and j-th row if $n_{i,j} < i$, and in the i-th column and $j+1$-th row otherwise. The matrix $\mathbf{I}_{6\times6}$ is always in the j-th column and row where $n_{i,j} = i$.

To complete the measurement update, we must compute the measurement covariance matrix as well as the transformed data vector required for the information filter. These are given by,

$$\overline{U}_i = \overline{H_i^\mathsf{T} R_i}^{-1} \overline{H_i} \tag{57}$$

$$u_i = \overline{H_i^\mathsf{T} R_i}^{-1} z_i \tag{58}$$

$$U_i = H_i^\mathsf{T} R_i^{-1} H_i, \tag{59}$$

where,

$$\overline{R}_i := \mathrm{Cov}[v_i] = \mathrm{blkdiag}\left\{\overline{R_{n_{i,m},i}}\right\}_{m=1}^{k_i+1} \tag{60}$$

$$R_i := \mathrm{Cov}[v_i] = \mathrm{blkdiag}\left\{R_{n_{i,m},i}\right\}_{m=1}^{k_i+1}. \tag{61}$$

Here, $\overline{R_{n_{i,m},i}} \in \mathbb{R}^{6\times6}$ is the covariance of the measurement of the pose of satellite m as taken by satellite i in Equation (52). The matrix $R_{n_{i,m},i}$ is the 'extension' of $\overline{R_{n_{i,m},i}}$ to $\mathbb{R}^{8\times8}$, where a row and column are added corresponding to each scalar entry of the dual quaternion. We add entries to the diagonal (say, 1) in these new rows and columns such that the matrix is not singular, however, what is set in those two added entries is of no importance as those entries will not be propagated. Further note that $\overline{R}_i \in \mathbb{R}^{6(k_i+1)\times6(k_i+1)}$ and $R_i \in \mathbb{R}^{8(k_i+1)\times8(k_i+1)}$.

Next, the Kalman state update is computed by each node $i \in V$ via,

$$\Delta x_i = \begin{bmatrix} \overline{\Delta\delta\hat{\mathbf{q}}_{j_{i,1}/I}}^T & \overline{\Delta\delta\hat{\mathbf{b}}_{\omega_{j_{i,1}}}}^T & \cdots & \overline{\Delta\delta\hat{\mathbf{q}}_{j_{i,k_i+1}/I}}^T & \overline{\Delta\delta\hat{\mathbf{b}}_{\omega_{j_{i,k_i+1}}}}^T \end{bmatrix} = M_i\left(\overline{u_i} - \overline{(U_i\mathbf{1})}\right), \tag{62}$$

where M_i is given by Eq. (40) as,

$$M_i = \left(P_i^{-1} + \overline{U}_i\right)^{-1}. \tag{63}$$

Finally, the Kalman state update is used to compute the time update. Similarly as to the non-cooperative case in the supplementary material [33], each node $i \in V$ updates each dual quaternion entry in Δx_i for $k = 1, \ldots k_i + 1$ as,

$$\Delta\delta\hat{q}_{j_{i,k},r} = \left(\sqrt{1 - ||\overline{\Delta\delta\hat{q}_{j_{i,k},r}}||^2}, \overline{\Delta\delta\hat{q}_{j_{i,k},r}}\right)$$

$$\Delta\delta\hat{q}_{j_{i,k},d} = \left(\frac{-\overline{\Delta\delta\hat{q}_{j_{i,k},r}}^T \overline{\Delta\delta\hat{q}_{j_{i,k},d}}}{\sqrt{1 - ||\overline{\Delta\delta\hat{q}_{j_{i,k},r}}||^2}}, \overline{\Delta\delta\hat{q}_{j_{i,k},d}}\right) \tag{64}$$

and again each node $i \in V$ concludes the measurement update by computing the estimate for $k = 1, \ldots k_i + 1$ as,

$$\hat{\mathbf{q}}_{i,k/I}^+ = \hat{\mathbf{q}}_{i,k/I}^- \Delta\delta\hat{\mathbf{q}}_{j_{i,k/I}}$$

$$\overline{\hat{\mathbf{b}}_{\omega_{i,k}}^+} = \overline{\hat{\mathbf{b}}_{\omega_{i,k}}^-} + \overline{\Delta\delta\hat{\mathbf{b}}_{\omega_{i,k}}}. \tag{65}$$

The DDQ-MEKF is summarized in Algorithm 1.

C. Consensus

In Algorithm 1, the satellites communicate with each other for velocity measurements only. In this subsection, we describe a consensus algorithm that allows each satellite to update and improve its state estimate by incorporating the state estimates of its neighbours.

We propose two consensus algorithms, denoted *soft* and *hard*. The goal of soft consensus is to average the state estimates of neighbouring satellites without any knowledge of the error covariance. Hard consensus, building upon [27], fuses covariance data exchanged between neighbours.

1. Description of the Soft Consensus Term

Before formally stating the algorithm, we give an example which illustrates the soft consensus step. Consider satellite 4 in Fig. 2. Satellite 4 has two neighbours, namely satellites 1 and 3, and thus has no notion of any other satellites in the fleet. After the measurement update of each satellite, the available estimates are listed in Table 1. Hence,

---

Algorithm 1 Distributed Dual Quaternion Kalman Filter

Require: $\forall i \in J : Q_i, R_i, \overrightarrow{\hat{q}}_i(t_0) := q_0, \hat{P}_i(t_0) := P_0$

  while $t_k \geq 0$ **do**

    for $i \in J$ **do**

      Take measurement $\omega_{B_i/I,m}^{B_i}$

      Propagate measurement to neighbours $j \in N_i$

      Compute dual velocity estimate $\overrightarrow{\hat{\omega}}_i = \overrightarrow{\omega}_{i,m} - \overrightarrow{\hat{b}}_i$

      Propagate each estimate $\overrightarrow{\hat{q}_i^+} \rightarrow \overrightarrow{\hat{q}_i^-}$ via Eqn. (46)

      Compute F, G; propagate P via Eqns. (48), (3)

      Measure own and relative poses z_i via Eqn. (52)

      Compute $\overline{H_i}$ and H_i matrix via Eqn. (56)

      Compute $\overline{U_i}, U_i$, and $\overline{u_i}$ via Eqn. (57), (58) & (59)

      Compute M_i via Eqn. (63)

      Compute Δx_i via Eqn. (62)

      Extend Kalman state update via Eqn. (64)

      Update estimate and bias via Eqn. (65)

    end for

  end while

Satellite 1	Satellite 3	Satellite 4

$$\overrightarrow{\hat{\mathbf{q}}_1^+} = \begin{bmatrix} \hat{\mathbf{q}}_{1,1/I}^+ \\ \hat{\mathbf{q}}_{1,2/I}^+ \\ \hat{\mathbf{q}}_{1,4/I}^+ \end{bmatrix} \qquad \overrightarrow{\hat{\mathbf{q}}_3^+} = \begin{bmatrix} \hat{\mathbf{q}}_{3,2/I}^+ \\ \hat{\mathbf{q}}_{3,3/I}^+ \\ \hat{\mathbf{q}}_{3,4/I}^+ \\ \hat{\mathbf{q}}_{3,5/I}^+ \end{bmatrix} \qquad \overrightarrow{\hat{\mathbf{q}}_4^+} = \begin{bmatrix} \hat{\mathbf{q}}_{4,1/I}^+ \\ \hat{\mathbf{q}}_{4,3/I}^+ \\ \hat{\mathbf{q}}_{4,4/I}^+ \end{bmatrix}$$

Table 1 Example of available estimates to each satellite, following Fig. 2.

satellite 4 can use certain measurements from its neighbour to improve its own state estimates. From satellite 1, it can use $\hat{\mathbf{q}}_{1,1}^+$ and $\hat{\mathbf{q}}_{1,4}^+$; similarly, from satellite 3 it can use $\hat{\mathbf{q}}_{3,3}^+$ and $\hat{\mathbf{q}}_{3,4}^+$. The other estimates from the neighbouring satellites are not used, as those are estimates of states of satellites not in the neighbourhood of satellite 4. The same notion pertains to the estimated biases.

The idea is now to average the estimated states of satellite 4, namely $\overrightarrow{\hat{q}_4^+}$, with the estimated states of satellite 1 and 3 which they have in common. In other words we would like to implement the consensus term in Eq. (41),

$$\phi_i = \mu_i \sum_{j \in N_i} \left(x_j^- - x_i^- \right), \tag{66}$$

adapted to the dual quaternion setting. As our state estimate is the *error* of the quaternion, it is nonsensical to use Eq. (66). Instead, the Kalman state update is computed first, then the pose estimate is updated, and then a consensus step is performed. Furthermore, computing the arithmetic difference of the pose as is done in Eq. (66) must be adapted to the dual quaternion setting; hence, we propose a consensus step which utilizes a more natural notion of averaging dual quaternions.

Soft Consensus on the Position and Velocity:
After performing the measurement update, each satellite $i \in V$ has an estimate of its own pose and bias and the pose and bias of its neighbours:

$$\overrightarrow{\hat{\mathbf{q}}_i^+} = \begin{bmatrix} \hat{\mathbf{q}}_{i,j_{i,1}/I}^{+,T} & \cdots & \hat{\mathbf{q}}_{i,j_{i,k_i+1}/I}^{+,T} \end{bmatrix}^T \in \mathbb{R}^{8(k_i+1)} \tag{67}$$

$$\overrightarrow{\hat{b}_i^+} = \begin{bmatrix} \overline{\hat{\mathbf{b}}_\omega^{+,T}}_{i,j_{i,1}} & \cdots & \overline{\hat{\mathbf{b}}_\omega^{+,T}}_{i,j_{i,1}} \end{bmatrix}^T \in \mathbb{R}^{6(k_i+1)}. \tag{68}$$

Each satellite $i \in V$ first needs to propagate this data to its neighbours $k \in N_i$:

$$\text{msg}_{i,k} = \left(\overrightarrow{\hat{\mathbf{q}}_i^+}, \overrightarrow{\hat{b}_i^+} \right). \tag{69}$$

We now focus on the issue of the correct arithmetic notion of the 'difference' between two dual quaterions. First, the dual part of the dual quaternion is transformed to the position using $r_{B/I}^B = \begin{bmatrix} 0 & \overline{r_{B/I}^B}^T \end{bmatrix}^T = 2p_{B/I}q_{B/I}$, yielding the position estimate for each $i \in V$,

$$\overline{\hat{r}_i^+} = \begin{bmatrix} \overline{\hat{r}_{i,j_{i,1}}^{+,T}} & \cdots & \overline{\hat{r}_{i,j_{i,k_i+1}}^{+,T}} \end{bmatrix}^T \in \mathbb{R}^{3(k_i+1)}. \tag{70}$$

Since each satellite is estimating its own state and that of its neighbours, the various state vectors of each satellite are not compatible with one another. To rectify this, we extend each satellite's estimated position vector to $\mathbb{R}^{3|V|} = \mathbb{R}^{3l}$ by padding the entries of untracked satellites with zeros. Then, we copy this position vector for each neighbour $k \in N_i$ of satellite i, and delete position estimates that satellite i and k do not have in common.

Formally, we define the subset $\Lambda_{i,k} := (V_i \cap V_k) \subset V$, in other words the set of nodes which satellite i and k have in common. For example, satellite 2 and 3 in the running example of Fig. 2 have $\Lambda_{2,3} = \{2, 3\}$. We write $\Lambda_{i,k} = \{\lambda_{ik,1}, \ldots, \lambda_{ik,|\Lambda_{i,k}|}\} \subset V$, i.e. $\Lambda_{i,k}$ is sorted in ascending order, as again, we assume that satellites know their own label in V. The full position estimate vector of satellite $i \in V$ with respect to its neighbour $k \in N_i$ is thus defined as,

$$\overrightarrow{\hat{r}_{\text{full},i,k}^+} = \begin{bmatrix} \overline{r_1}^T & \cdots & \overline{r_l}^T \end{bmatrix}^T \in \mathbb{R}^{3l} \tag{71}$$

$$\text{where } \overline{r_j} = \begin{cases} \overline{\hat{r}_{i,j}^+}, & \text{if } j \in \Lambda_{i,k} \\ \mathbf{0}_{3\times 1}, & \text{otherwise,} \end{cases} \tag{72}$$

and where $r_{i,j}^+$ is the estimated position of satellite j by satellite i.

We can now easily subtract position vectors between neighbouring satellites to compute their difference:

$$\phi_{\text{full},i,r} = \mu_{i,r} \sum_{j \in N_i} \left(\overrightarrow{\hat{r}_{\text{full},j,i}^+} - \overrightarrow{\hat{r}_{\text{full},i,j}^+} \right) \in \mathbb{R}^{3l}. \tag{73}$$

Now, $\phi_{\text{full},i,r}$ needs to be reduced to remove the padded zeros corresponding to positions that satellite i does not estimate. This is achieved by removing each entry corresponding to the indices of $\overline{r_j}$ in Eq. (71) which satisfies $j \notin V_i$, yielding $\phi_{i,r} \in \mathbb{R}^{3(k_i+1)}$. This concludes the portion of the soft consensus step on the position part of the dual quaternion.

14

For velocity consensus, as one can see from Eq. (45) it is sufficient to perform consensus on the bias. Therefore, achieving consensus on the bias automatically achieves consensus on the velocity. Consensus on the bias is performed using the same procedure as on position. We define the full bias vector of satellite $i \in V$ with respect to its neighbour $k \in N_i$ as,

$$\overrightarrow{\hat{b}^+}_{\text{full},i,k} = \begin{bmatrix} b_1^T & \cdots & b_l^T \end{bmatrix}^T \in \mathbb{R}^{6l} \tag{74}$$

$$\text{where } b_j = \begin{cases} \overrightarrow{\hat{\mathbf{b}}^+}_{\omega i,j}, & \text{if } j \in \Lambda_{i,k} \\ \mathbf{0}_{6\times 1}, & \text{otherwise,} \end{cases} \tag{75}$$

and where $\overrightarrow{\hat{\mathbf{b}}^+}_{\omega i,j}$ is the estimated bias of satellite j by satellite i. The consensus step on the bias is therefore,

$$\phi_{\text{full},i,b} = \mu_{i,b} \sum_{j \in N_i} \left(\overrightarrow{\hat{b}^+}_{\text{full},j,i} - \overrightarrow{\hat{b}^+}_{\text{full},i,j} \right) \in \mathbb{R}^{6l}. \tag{76}$$

The resulting vector $\phi_{\text{full},i,b}$ is then reduced by removing the extraneous entries corresponding to the padded zeros, as was done on $\phi_{\text{full},i,r}$, yielding $\phi_{i,b} \in \mathbb{R}^{6(k_i+1)}$.

Soft Consensus on the Attitude

Next, we define consensus on the attitude quaternion by exploiting a notion of 'averaging' a quaternion. First, for each $i \in V$, we define the attitude vector as,

$$\overrightarrow{\hat{q}_i^+} = \begin{bmatrix} \hat{q}_{i,j_{i,1}}^{+,T} & \cdots & \hat{q}_{i,j_{i,k_i+1}}^{+,T} \end{bmatrix}^T \in \mathbb{R}^{4(k_i+1)}. \tag{77}$$

Next, as for the position vector case, we pad the attitude vector with additional elements for the parts of the attitude corresponding to satellites which i doesn't track, and create copies of the vector for each neighbour $k \in N_i$ which only contain estimates in common:

$$\overrightarrow{\hat{q}^+}_{\text{full},i,k} = \begin{bmatrix} q_1^{i,k,T} & \cdots & q_l^{i,k,T} \end{bmatrix}^T \in \mathbb{R}^{4l} \tag{78}$$

$$\text{where } q_j^{i,k} = \begin{cases} \hat{q}_{i,j}^+, & \text{if } j \in \Lambda_{i,k} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T, & \text{otherwise,} \end{cases} \tag{79}$$

and where $q_{i,j}^+$ is the estimated attitude of satellite j by satellite i.

The correct notion of the 'difference' between quaternions q_1 and q_2 can be expressed as $q_1^* q_2$, and the correct notion of 'summation' over all of the quantites is given by quaternion multiplication. Hence, the correct notion of the consensus step for the attitude quaternions is given by,

$$\theta_{\text{full},i,q} = \prod_{j \in N_i} \left(\overrightarrow{\hat{q}^+}_{\text{full},i,j} \right)^* \left(\overrightarrow{\hat{q}^+}_{\text{full},j,i} \right) \tag{80}$$

$$= \prod_{j \in N_i} \begin{bmatrix} \left(q_1^{i,j} \right)^* \left(q_1^{j,i} \right) \\ \vdots \\ \left(q_l^{i,j} \right)^* \left(q_l^{j,i} \right) \end{bmatrix} = \begin{bmatrix} \theta_{i,q_1} \\ \vdots \\ \theta_{i,q_l} \end{bmatrix} \in \mathbb{R}^{4l}, \tag{81}$$

where the product shown here can be understood as a quaternion multiplication which acts on each quaternion in the vector independently.

The last step needed for the soft consensus is to correctly define a tuning constant μ, which acts as a feedback gain on the quaternion consensus term in Eq. (81). To preserve the normalization of the quaternion, we define the quaternion scaling function,

$$q_s(q,\mu) = \begin{bmatrix} \sqrt{1 - \mu^2 \sum_{k=1}^3 q_k^2} \\ \mu q_1 \\ \mu q_2 \\ \mu q_3 \end{bmatrix}, \tag{82}$$

15

and with this function the quaternion error is scaled appropriately as,

$$\phi_{\text{full},i,q}^{\mu} := \begin{bmatrix} q_s(\theta_{i,q_1}, \mu_{i,q}) \\ \vdots \\ q_s(\theta_{i,q_l}, \mu_{i,q}) \end{bmatrix} \in \mathbb{R}^{4l}. \tag{83}$$

As a consistency check, as $\mu \to 0$, the attitude approaches $[1\ 0\ 0\ 0]$, which denotes an error of zero. Finally, as for the bias and position vectors, the last step is to reduce this quantity to the appropriate size by only keeping the entries which are in the set V_i, yielding $\phi_{i,q} \in \mathbb{R}^{4(k_i+1)}$.

Combining the Soft Consensus Terms

Now that the three soft consensus terms are computed, they need to be applied to the estimate of satellite i. To apply the correction term on the attitude, the current estimate of the attitude is multiplied (in the quaternion sense) with the attitude component of the soft consensus term $\phi_{i,q}$ as,

$$\overline{\hat{q}_i^{++}} = \overline{\hat{q}_i^{+}}\phi_{i,q} = \begin{bmatrix} \hat{q}_{i,j_{i,1}}^{+}\phi_{i,q_1} \\ \vdots \\ \hat{q}_{i,j_{i,k_i+1}}^{+}\phi_{i,q_{k_i+1}} \end{bmatrix}. \tag{84}$$

The bias estimate is corrected by adding the bias soft consensus term as,

$$\overrightarrow{\hat{b}_i^{++}} = \overrightarrow{\hat{b}_i^{+}} + \phi_{i,b}. \tag{85}$$

Finally, the position soft consensus term is added to Eq. (70)

$$\overrightarrow{\hat{r}_i^{++}} = \overrightarrow{\hat{r}_i^{+}} + \phi_{i,r}, \tag{86}$$

and then transformed to the dual quaternion position, as

$$\overrightarrow{\hat{q}_i^{++}} = \begin{bmatrix} \hat{q}_{i,j_{i,1}}^{++} \\ \frac{1}{2}\begin{bmatrix} 0 \\ \hat{r}_{i,j_{i,1}}^{++} \end{bmatrix}\hat{q}_{i,j_{i,1}}^{++} \\ \vdots \\ \hat{q}_{i,j_{i,k_i+1}}^{++} \\ \frac{1}{2}\begin{bmatrix} 0 \\ \hat{r}_{i,j_{i,k_i+1}}^{++} \end{bmatrix}\hat{q}_{i,j_{i,k_i+1}}^{++} \end{bmatrix}. \tag{87}$$

2. Description of the Hard Consensus Term

The purpose of the hard consensus term is to fuse measurement data and error covariance matrices between neighbouring agents. We adapt the method proposed in [27] to the dual quaternion setting. The main challenge is computing the distributed versions of the Kalman information filter quantities in (13), as each satellite has a different number of estimated states and so the measurement matrices have different sizes for each node and also incorporate different measurements. As such, each satellite needs to prepare a measurement matrix and vector compatible with each of its neighbours.

Consider the measurement matrix of satellite i as defined in Eq. (56). Then, $H_{i,i} = H_i \in \mathbb{R}^{8(k_i+1)\times14(k_i+1)}$. We will define the matrix $H_{i,k}$ as just a reconfiguration of $H_{i,i}$ to match the different state of satellite k, as shown in Appendix A.B Eqs. (140)-(141). Because some states are not estimated by satellite i but are estimated by its neighbour satellite k, it may be the case that depending on the neighbourhood, some of the measurement Jacobians satisfy $H_{i,n_{k,1}}^{n_{k,1}} = \mathbf{0}_{8\times8}$.

A similar procedure can now be done for the measurement itself. As for the measurement Jacobian, each satellite needs to prepare the measurement and convert it to a vector compatible with the state estimate of the neighbouring

satellites. The measurement of satellite i is defined as in Eq. (52). Then, the vector $v_o \in \mathbb{R}^6$ (being the o-th entry of $z_{i,k}$) is equal to the vector $c_l \in \mathbb{R}^6$ (being the l-th entry of z_i) if $j_{i,l} = j_{k,o}$. Otherwise the entry is $\mathbf{0}_{6\times1}$. Explicitly, for $k \in V_i$,

$$z_{i,k} = \begin{bmatrix} v_1 \\ \vdots \\ v_{k_k+1} \end{bmatrix} \quad \text{where} \quad \begin{cases} v_o = c_l, & \text{if } j_{i,l} = j_{k,o} \\ v_o = \mathbf{0}_{6\times1}, & \text{otherwise} \end{cases}. \tag{88}$$

In the running example described in Fig 5, the sensitivity matrices and measurement vectors of satellite 4 for the estimates of itself and its neighbours 1 and 3 are shown in Eqns (89a), (89b) and (89c).

$$\overline{H_{4,1}} = \begin{bmatrix} \overline{H_{4,1}^1} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \overline{H_{4,1}^4} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{I}_{6\times6} & \mathbf{0}_{6\times6} \end{bmatrix} \in \mathbb{R}^{18\times36} \qquad z_{4,1} = \begin{bmatrix} \overline{q}_{B_4/B_1,m} \\ r^{B_4}_{B_1/B_4,m} \\ \mathbf{0}_{6\times1} \\ \hat{\mathbf{q}}_{4,4/I}\mathbf{q}_{B_4/I,m} \end{bmatrix} \tag{89a}$$

$$\overline{H_{4,3}} = \begin{bmatrix} \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \overline{H_{4,3}^3} & \mathbf{0}_{6\times6} & \overline{H_{4,3}^4} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{I}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \end{bmatrix} \in \mathbb{R}^{24\times48} \qquad z_{4,3} = \begin{bmatrix} \mathbf{0}_{6\times1} \\ \overline{q}_{B_4/B_3,m} \\ r^{B_4}_{B_3/B_4,m} \\ \hat{\mathbf{q}}_{4,4/I}\mathbf{q}_{B_4/I,m} \\ \mathbf{0}_{6\times1} \end{bmatrix} \tag{89b}$$

$$\overline{H_{4,4}} = \begin{bmatrix} \overline{H_{4,1}^1} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \overline{H_{4,1}^1} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \overline{H_{4,3}^3} & \mathbf{0}_{6\times6} & \overline{H_{4,3}^4} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{I}_{6\times6} & \mathbf{0}_{6\times6} \end{bmatrix} \in \mathbb{R}^{18\times36} \qquad z_{4,4} = \begin{bmatrix} \overline{q}_{B_4/B_1,m} \\ r^{B_4}_{B_1/B_4,m} \\ \overline{q}_{B_4/B_3,m} \\ r^{B_4}_{B_3/B_4,m} \\ \hat{\mathbf{q}}_{4,4/I}\mathbf{q}_{B_4/I,m} \\ \mathbf{0}_{6\times1} \end{bmatrix} \tag{89c}$$

Now that the measurement matrices and vectors are compatible between neighbours, we can formalize the consensus step. The only equations which change in comparison to the distributed Kalman filter without a consensus step are Eqs. (57)-(59). Each $i \in V$ satellite prepares the measurement covariance matrix and the measurement for its neighbour $k \in N_i$,

$$\begin{aligned} \overline{U_{i,k}} &= \overline{H_{i,k}^T} R_{i,k}^{-1} \overline{H_{i,k}} \\ U_{i,k} &= H_{i,k}^T R_{i,k}^{-1} H_{i,k} \\ u_{i,k} &= \overline{H_{i,k}^T} R_{i,k}^{-1} \overline{z_{i,k}}, \end{aligned} \tag{90}$$

and sends a message to its neighbour k,

$$\text{msg}_{i,k} = \left(\overline{U_{i,k}}, U_{i,k}, u_{i,k} \right). \tag{91}$$

Next, each satellite $i \in V$ calculates the aggregated covariance matrix S_i and the aggregated sensor data y_i as,

$$y_i = \sum_{j\in V_i} u_{j,i} \qquad S_i = \sum_{j\in V_i} U_{j,i} \qquad \overline{S_i} = \sum_{j\in V_i} \overline{U_{j,i}}. \tag{92}$$

Each satellite $i \in V$ then computes the state update (63) as,

$$\Delta x_i = \begin{bmatrix} \overline{\Delta\delta\hat{\mathbf{q}}_{j_{i,1}/I}^T} & \overline{\Delta\delta\hat{\mathbf{b}}_{\omega_{j_{i,1}}}^T} & \cdots & \overline{\Delta\delta\hat{\mathbf{q}}_{j_{i,k_i+1}/I}^T} & \overline{\Delta\delta\hat{\mathbf{b}}_{\omega_{j_{i,k_i+1}}}^T} \end{bmatrix}^T = M_i \left(\overline{y_i} - \overline{(S_i\mathbf{1})} \right), \tag{93}$$

where

$$M_i = \left(P_i^{-1} + \overline{S_i} \right)^{-1}. \tag{94}$$

Finally, each agent $i \in V$ updates the estimates of the pose and bias using Eqs. (64) and (65).

D. Leader-Follower Consensus

In a satellite fleet where absolute pose measurements are difficult to obtain or would require an expensive sensor (i.e., a star-tracker), it might be interesting to only provide pose measurements to a subset of the satellite fleet. The remaining satellites would then only have relative pose measurements with their neighbours.

In this configuration, we define a set of *leaders* $L \subset V$ which includes all satellites that have access to absolute pose measurements. We then denote $F = V \setminus L$ as the *followers*.

The removal of the pose measurement for the followers only changes their respective measurement update of the Kalman filter. A straightforward modification consists of removing the absolute pose measurement from the measurement vector and remove the corresponding row of the measurement matrix in (56) for all satellites $i \in F$. Empirically, this solution does not yield good results if the set L is small. Hence, a different approach was chosen for the estimation of the followers.

1. Calculation of Absolute Pose Measurements

To address this issue, we propose a method whereby an absolute pose measurement is constructed from relative pose measurements and absolute pose estimates. At each timestep, satellite $i \in V$ estimates its own pose and the pose of its neighbours $k \in N_i$. Additionally, it measures the relative pose of the neighbouring satellites in its own body frame. Satellite i can thus compute $|N_i| = k_i$ absolute pose measurements for $k = 1, \ldots, k_i$ through

$$q_{B_i/I,m_k} = \hat{q}^-_{i,n_{i,k}} q^*_{B_{n_{i,k}}/B_i,m} \tag{95}$$

$$r^I_{B_i/I,m_k} = 2\hat{q}^-_{i,n_{i,k}} \hat{p}^-_{i,n_{i,k}} - q_{B_{n_{i,k}}/B_i,m} r^{B_i}_{B_{n_{i,k}}/B_i,m} q^*_{B_{n_{i,k}}/B_i,m}. \tag{96}$$

Informally, to get an absolute attitude measurement of satellite i, we remove the relative attitude measurement from satellite i to satellite k from the estimated attitude of satellite k computed by satellite i. The same is done for the position measurement. To fuse the various position estimates, we can average them as,

$$\overline{r^I_{B_i/I,m}} = \frac{1}{k_i} \sum_{k \in N_i} \overline{r^I_{B_i/I,m_k}}. \tag{97}$$

The notion of the 'average' of the attitude quaternion is proposed by [38] and is computed as the minimizer of the linear program,

$$\overline{q}_{B_i/I,m} = \begin{cases} \operatorname{argmax} & \operatorname{Tr}\left(A(q_{B_i/I,m})C^T\right) \\ \text{subject to} & C = \sum_{k \in N_i} A(q_{B_i/I,m_k}), \end{cases} \tag{98}$$

where $A(q)$ is the rotation matrix of quaternion q. The dual quaternion for the pose measurement is then constructed from $\overline{r^I_{B_i/I,m}}$ and $\overline{q}_{B_i/I,m}$, and no further changes to the algorithm are necessary.

2. Stubborn Leaders

In the leader-follower setting, it is quite clear that leaders always have more information than followers due to their absolute pose measurement. In many cases, a follower may only be able to improve the estimate of a leader if it has immediate knowledge of another leader satellite, i.e., if the follower is connected to two or more leaders. Notwithstanding pathological cases, this occurs in a graph when a large percentage of satellites are leaders. Conversely, as the percentage of leaders becomes small, the follower satellites may very well reduce the estimator performance of the leader, which in turn may deteriorate the performance of the whole fleet. In this section, we describe the notion of a 'stubborn' leader which does not perform consensus with followers, which, as we will see in §IV.C, provides better average performance when the percentage of leaders is low. Only two changes need to be made. In the case of soft consensus, it suffices to set $\mu_{i,r} = \mu_{i,q} = \mu_{i,b} = 0$ if $i \in L$ for leaders to ignore the follower estimates. In the case of hard consensus, Eq. (92) must be replaced with,

$$y_i = \begin{cases} u_{i,i}, & \text{if } i \in L \\ \sum_{j \in V_i} u_{j,i}, & \text{otherwise,} \end{cases} \quad S_i = \begin{cases} U_{i,i}, & \text{if } i \in L \\ \sum_{j \in V_i} U_{j,i}, & \text{otherwise,} \end{cases} \quad \overline{S}_i = \begin{cases} \overline{U_{i,i}}, & \text{if } i \in L \\ \sum_{j \in V_i} \overline{U_{j,i}}, & \text{otherwise.} \end{cases} \tag{99}$$

With these modifications, the leaders will not be influenced by the follower estimates.

IV. Numerical Examples

In this section, we present three numerical experiments that demonstrate the advantages of the distributed DQ-MEKF. First, in §IV.A we compare the DQ-MEKF and its distributed counterpart. Next, in §IV.B we showcase the filter in a scenario where a fleet of satellites swarm around an asteroid. Finally, in §IV.C, we examine the performance of the distributed DQ-MEKF in the leader-follower scenario where only a subset of the satellites have an absolute pose measurement. We provide an open-source repository [32] containing the code to reproduce the numerical examples in this section.

A. Comparison of Cooperative vs Non-Cooperative Fleet

Matrix	Value
$\overline{Q}_{b_\omega}$	$\frac{10^{-3}}{\text{SNR}^2}\mathbf{I}_{3\times3}$
$\overline{Q}_{b_v}$	$\frac{10^{-1}}{\text{SNR}^2}\mathbf{I}_{3\times3}$
$\overline{Q}_\omega, \overline{Q}_v$	$\mathbf{0}_{3\times3}$
$R_{6\times6}$	$\text{blkdiag}\{\text{std}_q^2\mathbf{I}_{3\times3}, \text{std}_r^2\mathbf{I}_{3\times3}\}$
$P_{12\times12}(0)$	$\text{blkdiag}\{10^{-1}\mathbf{I}_{6\times6}, 10^{-2}\mathbf{I}_{6\times6}\}$
$P_{15\times15}(0)$	$\text{blkdiag}\{10^{-1}\mathbf{I}_{6\times6}, 10^{-2}\mathbf{I}_{9\times9}\}$

Table 2   Simulation parameters for the comparison between the DQ-MEKF and DDQ-MEKF.

We aim to compare the performance of Algorithm 1 on a fleet of satellites with that same fleet of satellites running the DQ-MEKF algorithm from [19], as this essentially quantifies the improvement of performance when distributing the algorithm.

In particular, we compare the performance of the DQ-MEKF, Algorithm 1 with both hard and soft consensus, and Algorithm 1 with soft consensus only. We assume absolute pose measurements for all satellites, and in each case, the respective filter was initialized with the parameters in Table 2. In the distributed case, for each satellite $i \in V$, the covariance matrices for the various noises are direct-summed with a copy for each neighbour $j \in V_i$. Whenever soft consensus is employed, we set $\mu_{i,r} = \mu_{i,q} = \mu_{i,b} = 1/|J_i|$ where $|J_i|$ is the number of neighbours of i, including itself.

Each simulation is initialized with a given signal-to-noise ratio (SNR), and we do a simulation sweep over values of the SNR from 5 to 10^5. In Table 2, std_q and std_r are the standard deviations of the Gaussian noise applied to the attitude and position respectively, which are calculated from the given SNR. For each algorithm and each SNR, 80 such simulations were run at 20Hz for 60s. Finally, the graphs defining the connectivity of the fleet of satellites were generated randomly, with the probability of an edge existing between $i, j \in V$ being set to 0.5. Disconnected graphs generated in this manner were discarded and re-initialized.

The results of the parameter sweep are depicted in Fig. 3, where we plot the interquartile ranges of attitude, position, angular velocity and linear velocity estimator error over the simulations for varying SNRs. For small SNRs, soft consensus is is superior to the non-cooperative case and performs similarly to the case with both soft and hard consensus; however, for higher SNRs, soft consensus provides an inferior velocity estimate compared to the other filters. However, the DDQ-MEKF with both hard and soft consensus outperforms the non-cooperative case in every metric over all SNRs.

B. Distributed Swarming Around an Asteroid

In this example, we consider the performance of Algorithm 1 (with both hard and soft consensus) in a scenario where 50 satellites arrange themselves around an asteroid, while maintaining an attitude representative of pointing a scientific payload at the asteroid. The initial conditions of the satellites are arranged on a plane 40m from the asteroid, and a Fibonacci lattice [39] is used to evenly space the target positions of the satellites on a sphere of radius 25m centered around the asteroid. The reference attitude is constructed from the current estimate of the satellite and the position of the asteroid, and an LQR controller is used to maneuver the fleet to their final positions and attitudes over ~200s. During the maneuver, the satellites perform Algorithm 1 across the connectivity graph seen in Fig. 5, with a sampling rate of 20 Hz, and initialized with the parameters in Table 3.

A representative satellite with node label 10 is selected at random, and is coloured in blue in Figs. 4 and 5. Leader nodes are depicted in red. The position and attitude (in quaternion form) of satellite 10 throughout the maneuver is shown in Fig. 6 , along with the estimates of satellite 10 as computed by Algorithm 1 by satellite 10 itself, and its
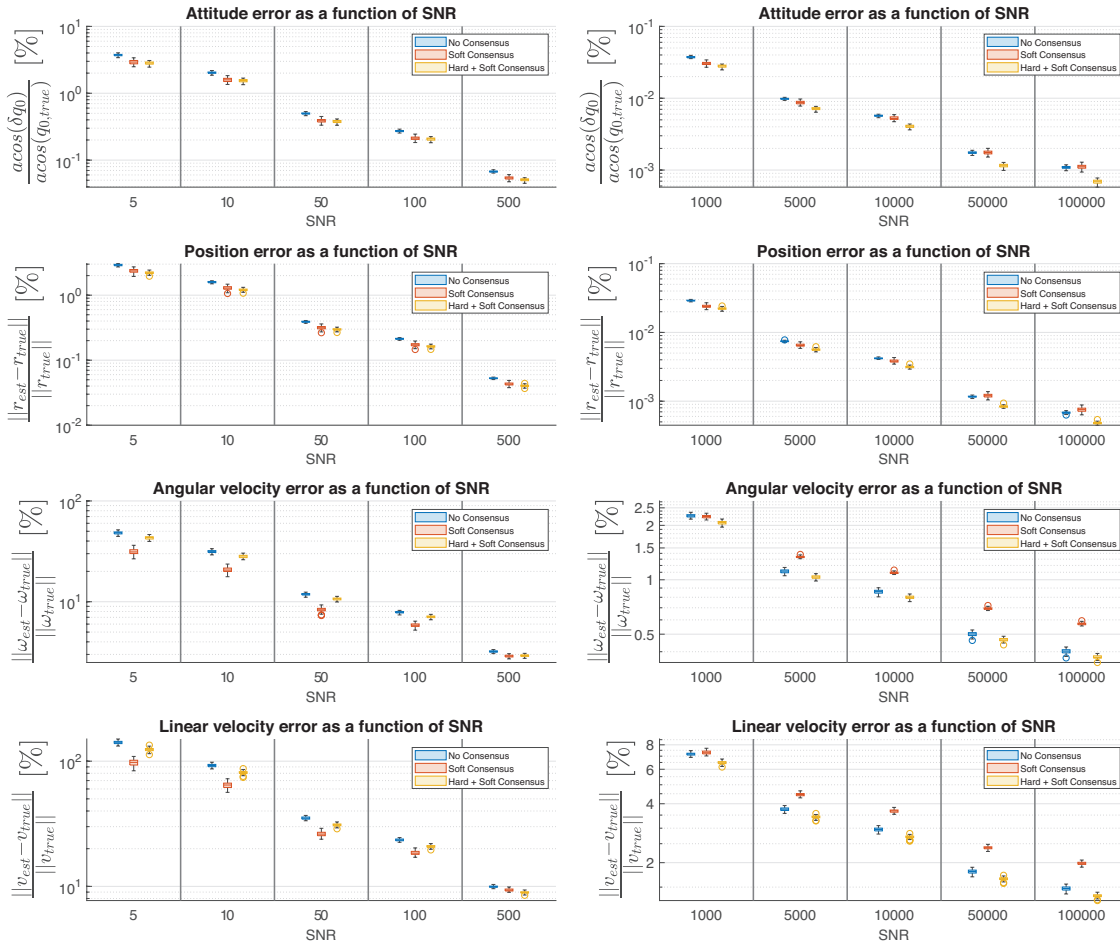
Fig. 3 **Comparison of the different consensus filters for a fleet of ten satellites. Blue: the baseline non-cooperative DQ-MEKF. Red: the distributed DQ-MEKF is shown with the soft consensus term. Yellow: the distributed DQ-MEKF with soft and hard consensus.**

neighbours. The RMS errors of the attitude, position, angular velocity, and linear velocity are shown in Fig. 7. As one can see, the error decreases over time until it reaches a steady-state, showing that the estimator remains stable during the maneuver. A supplementary video showing the swarming behaviour of entire the satellite fleet, as well as real time pose data from satellite 10, may be found at [40].

C. Leader-Follower Scenario

In this section, we provide a set of numerical experiments to investigate the performance of Algorithm 1 with leader-follower consensus as described in §III.D. Explicitly, we consider four cases: with hard and soft consensus, with soft consensus only, with hard and soft consensus and subborn leaders, and with soft consensus and stubborn leaders. In each case, only the leaders have access to absolute pose measurements, and followers create synthetic pose measurements using Eqn. (95).

For each experiment, the SNR was fixed to 1000, and the filters were initialized with the parameters in Table 2. We varied the percentage of leaders from 10%-100%, and leaders were selected uniformly at random in the graph in order to understand how the performance varies with different sets of leaders. The graph itself was randomized as in §IV.A. For each instance of leader percentage, the simulation was run 80 times (randomizing the graph and leaders each time),
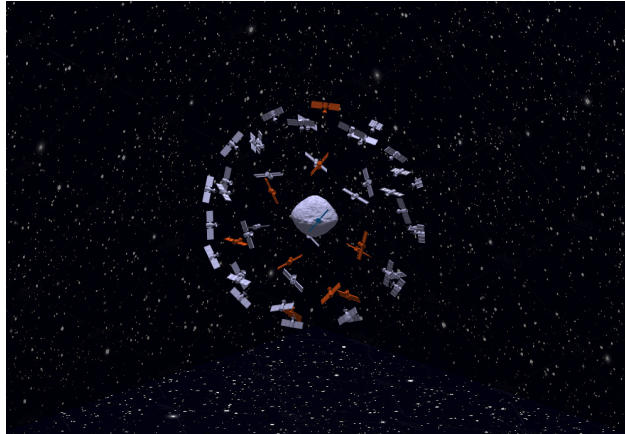
Fig. 4 Target formation of the cooperative asteroid swarming scenario. Satellites depicted in red are the leaders.
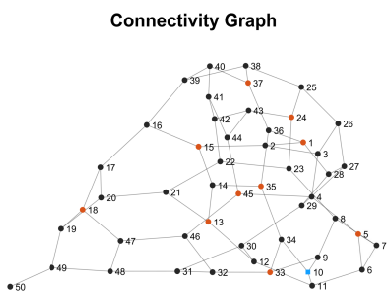
Fig. 5 Connectivity graph for the asteroid coverage scenario. Red nodes depict leaders, and the blue square node depicts satellite 10.
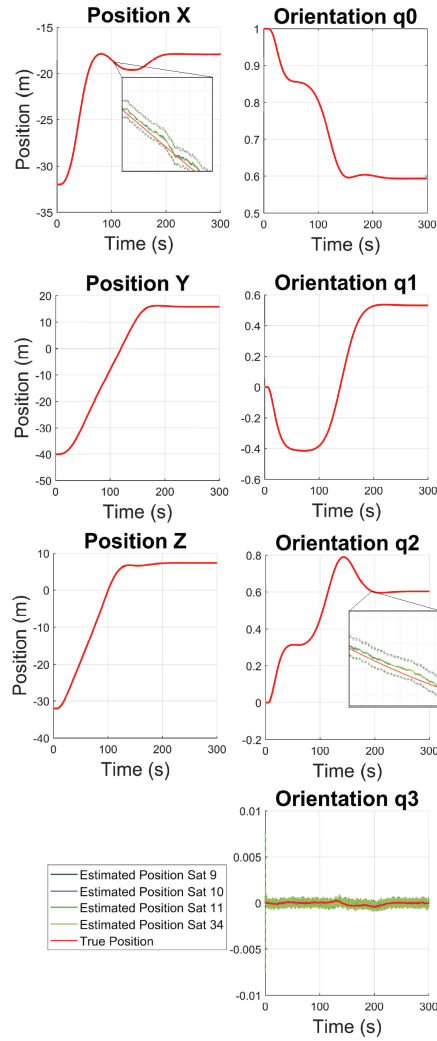
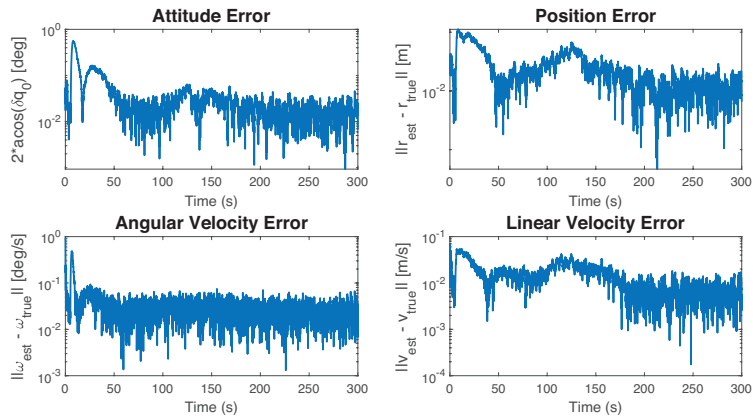**Fig. 6　Attitude and position of satellite 10 in the asteroid coverage scenario.**

Fig. 7　RMS error of the attitude and position and their derivatives of satellite 10 in the asteroid coverage scenario.

Parameter ($\forall i \in V$)	Value
Attitude Noise Cov	$2.79 \times 10^{-7}\mathbf{I}_{4\times4}$
Position Noise Cov	$8.55 \times 10^{-4}\mathbf{I}_{3\times3}$
$\overline{Q}_{\omega_i}$	blkdiag$\{0 \times \mathbf{I}_{6\times6}\}$
$\overline{Q}_{\mathbf{b}_{\omega_i}}$	blkdiag$\{10^{-6}\mathbf{I}_{3\times3}, 10^{-4}\mathbf{I}_{3\times3}\}$
$P_i(0)$	blkdiag$\{10^{-1}\mathbf{I}_{6\times6}, 10^{-2}\mathbf{I}_{6\times6}\}$
$\mu_{i,q}, \mu_{i,r}, \mu_{i,b}$	$1/|J_i|$
$Q_{LQR,i}$	$1 \times 10^{-1}\mathbf{I}_{12\times12}$
$R_{LQR,i}$	$1 \times 10^{-1}\mathbf{I}_{6\times6}$

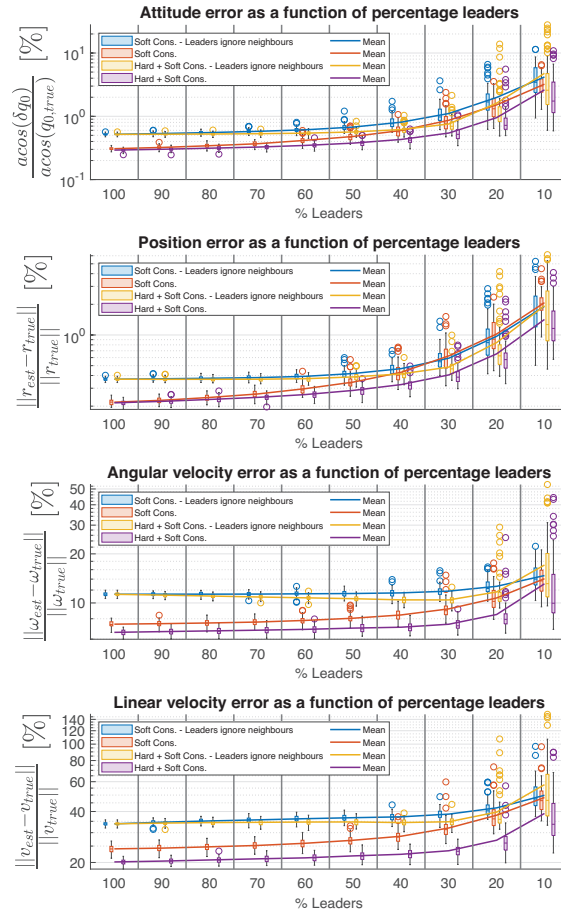**Table 3   Settings of the distributed DQ-MEKF for the distributed swarming example.**

Fig. 8   Comparison of the four variations of the DDQ-MEKF by varying the % of leaders over a random graph. The fleet has ten satellites, and has a fixed SNR of 1000.

and each simulation was run for a duration of 60s at 20Hz (1200 iterations). In order to understand the steady-state error of each simulation, the error values in Fig. 8 are computed using the last 600s of each simulation to discard transient behaviour.

We can see in Fig. 8 that algorithms with stubborn leaders perform worse than with non-stubborn leaders, and overall the algorithms perform better with larger percentages of leaders. This is expected. When leaders ignore their neighbours and all satellites are leaders, this is equivalent to the case in §IV.A with non-cooperating satellites, where the cooperating fleet has superior performance to that of the non-cooperating fleet. The soft consensus case also always does worse than the combination of hard and soft consensus for a SNR of 1000, which is in agreement with §IV.B.

As expected, as the percentage of leaders decreases, the performance of the entire fleet also decreases. It can also be seen that the case of stubborn leaders doesn't suffer as much performance degradation as the case where all satellites perform consensus. This is due to the fact that the followers cannot have a better estimate of the leader if the number of leaders is very small. A final observation is that the performance of the fleet doesn't degrade until approximately less than 50% of fleet are leaders. Since the graphs and leaders are randomly generated, this is the point at which a follower has a high likelihood of only being connected to one leader, or to no leaders at all. Therefore, its performance is limited by the quality of measurements it has access to, which in turn is determined in part by the quality of the estimates of its neighbours. If this neighbourhood doesn't contain leaders, then the quality of the corresponding satellite's estimate reflects the fact that the absolute pose measurements provided by the leaders is degraded as it propagates across the network.

V. Conclusion

In this paper, we have proposed, derived and validated a distributed version of the dual-quaternion multiplicative extended Kalman filter as studied by [19]. The relevant algebraic quantities for distributing the filter were derived, and the concepts of hard and soft consensus were introduced to improve measurement and estimate fusion between neighbouring satellites in the network. We also described the filter in a leader-follower scenario, where only leaders in the fleet have access to an absolute pose measurement.

Our numerical experiments show that distributing the DQ-MEKF over a network of satellites provides a substantial performance increase across a variety of signal-to-noise ratios over the non-cooperative scenario when both hard and soft consensus is employed alongside Algorithm 1. Furthermore, we established numerically that the filter is sufficiently stable and provides an estimate of sufficient fidelity that it can be succesfully used in a dynamic scenario where satellites are cooperating to swarm around an asteroid. This experiment also demonstated the scalability of the algorithm over a larger fleet. Our final numerical experiment showed that we can remove expensive absolute pose measurement sensors, such as star-trackers, from a large part of the fleet with no significant performance degradation. We leave the question of optimally choosing which satellites to select as leaders (given a network) for future research.

Additional future work may include validation of the proposed methods on a hardware testbed involving multiple satellite simulators, or adapting the filter to pose control of other vehicles such as quadrotors. Considering correlations between neighbouring pose estimates in the consensus algorithms may yield further improvements in performance. One remaining question not fully addressed in this paper is a detailed analysis of the effect of the network topology on the performance of the algorithm – our chosen network topologies were randomly generated, and as such the average effect of the networks were studied, but the question of optimizing specific networks for the filter remains. Aside from the dynamical analysis contained in this paper, this line of work will involve topics in combinatorial optimization.

A. Appendix

A. Jacobians of Measurement Models

Here, we derive the Jacobians of the measurement functions in (52) for various measurement models. For the sake of reducing notational burden, the time dependence of the estimates is neglected in the derivations below. However, we ask the reader to keep in mind that the estimates and measurements are also time-dependent.

1. Jacobian of the Absolute Pose Measurement

As can already be seen in Equation (52), the measurement is multiplied with the conjugate of the estimate, directly yielding the error dual quaternion. Hence, the measurement matrix H is simply,

$$H_{abs} = \begin{bmatrix} \mathbf{I}_{6\times6} & \mathbf{0}_{6\times6} \end{bmatrix}. \tag{100}$$

2. Jacobian of the Relative Attitude Measurement

The relative attitude q_r is the real part of $\mathbf{q}_{B_k/B_i} = q_{r,B_k/B_i} + \epsilon q_{d,B_k/B_I}$. To reduce notational burden, we will denote $q_{r,B_k/B_i} := q_{k/i}$ and $q_{d,B_k/B_I} := p_{k/i}$. In this way, $q_{k/i}$ is the quaternion describing the rotation from satellite i to satellite k, and $p_{k/i}$ denotes the position quaternion from satellite i to k.

We define the error quaternion as $\delta\mathbf{q}_{B/I} = \hat{\mathbf{q}}^*_{B/I}\mathbf{q}_{B/I}$, where B is the body frame of the satellite. Through the definition of the dual quaternion, it immediately follows that $\delta q_{B/I} = \hat{q}^*_{B/I}q_{B/I}$. Dropping B from the notation as above yields,

$$\delta q_{i/I} = \hat{q}^*_{i/I}q_{i/I} \tag{101}$$

$$\delta q_{k/I} = \hat{q}^*_{k/I}q_{k/I}, \tag{102}$$

which allows us to compute $h_{q_{k/i}}(\cdot)$ as,

$$q_{k/i} = q^*_{i/I}q_{k/I} \tag{103}$$

$$= (\hat{q}_{i/I}\delta q_{i/I})^*(\hat{q}_{k/I}\delta q_{k/I}) \tag{104}$$

$$=: h_{q_{k/i}}(x_i). \tag{105}$$

Given the measurement function $h_{q_{k/i}}(\cdot)$, we can now derive the corresponding Jacobian for the measured relative attitude by differentiating $q_{k/i}$ by the state variables $\delta q_{i/I}$ and $\delta q_{k/I}$, as

$$H_{q_{k/i},\delta q_{i/I}} = \left.\frac{\partial q_{k/i}}{\partial \delta q_{i/I}}\right|_{x_i=\hat{x}_i} \tag{106}$$

$$= \left.\frac{\partial((\hat{q}_{i/I}\delta q_{i/I})^*(\hat{q}_{k/I}\delta q_{k/I}))}{\partial \delta q_{i/I}}\right|_{x_i=\hat{x}_i} \tag{107}$$

$$= [|\hat{q}_{k/I}|]_R[|\hat{q}^*_{i/I}|]_R\frac{\partial(\delta q^*_{i/I})}{\partial \delta q_{i/I}} \tag{108}$$

$$= [|\hat{q}_{k/I}|]_R[|\hat{q}^*_{i/I}|]_R I^* \in \mathbb{R}^{4\times4}, \tag{109}$$

and,

$$H_{q_{k/i},\delta q_{k/I}} = \left.\frac{\partial q_{k/i}}{\partial \delta q_{k/I}}\right|_{x_i=\hat{x}_i} \tag{110}$$

$$= \frac{\partial((\hat{q}_{i/I}\delta q_{i/I})^*(\hat{q}_{k/I}\delta q_{k/I}))}{\partial \delta q_{k/I}} \tag{111}$$

$$= [|\hat{q}^*_{i/I}|]_L[|\hat{q}_{k/I}|]_L\frac{\partial(\delta q_{k/I})}{\partial \delta q_{k/I}} \tag{112}$$

$$= [|\hat{q}^*_{i/I}|]_L[|\hat{q}_{k/I}|]_L \in \mathbb{R}^{4\times4}. \tag{113}$$

3. Jacobian of the Relative Position Measurement

As before we will drop the subscript B denoting the body frame and introduce $q_{r,B_k/B_i} := q_{k/i}$ and $q_{d,B_k/B_I} := p_{k/i}$. Satellite i measures its position in its body frame relative to the frame of its neighbour k. Denote this measurement $r^i_{k/i}$. To compute the measurement function $h_{i,r_{ik}}(\cdot)$, we need two more equations, the position quaternion error and the transformation of the position onto the quaternion position. The former is computed as,

$$\delta\mathbf{q}_{B/I} = \hat{\mathbf{q}}^*_{B/I}\mathbf{q}_{B/I} \tag{114}$$

$$\mathbf{q}_{B/I} = \hat{\mathbf{q}}_{B/I}\delta\mathbf{q}_{B/I} \tag{115}$$

$$= (\hat{q}_{B/I}\delta q_{B/I}) + \epsilon(\hat{q}_{B/I}\delta p_{B/I} + \hat{p}_{B/I}\delta q_{B/I}) \tag{116}$$

$$= q_{B/I} + \epsilon p_{B/I} \tag{117}$$

Hence, $p_{B/I} = \hat{q}_{B/I}\delta p_{B/I} + \hat{p}_{B/I}\delta q_{B/I}$. The latter quantity is simply $r_{B/I}^B = \begin{bmatrix} 0 & \overline{r_{B/I}^B}^\top \end{bmatrix}^\top = 2p_{B/I}q_{B/I}$. We can thus compute the measurement function as,

$$r_{k/I}^I = r_{i/I}^I + q_{i/I}r_{k/i}^i q_{i/I}^* \tag{118}$$

$$r_{k/i}^i = q_{i/I}^*(r_{k/I}^I - r_{i/I}^I)q_{i/I} \tag{119}$$

$$= q_{i/I}^*(2p_{k/I}q_{k/I}^* - 2p_{i/I}q_{i/I}^*)q_{i/I} \tag{120}$$

$$= 2(\delta q_{i/I}^*\hat{q}_{i/I}^*)((\hat{p}_{k/I}\delta q_{k/I})(\delta q_{k/I}^*\hat{q}_{k/I}^*))(\hat{q}_{i/I}\delta q_{i/I}) \tag{121}$$

$$+ 2(\delta q_{i/I}^*\hat{q}_{i/I}^*)((\hat{q}_{k/I}\delta p_{k/I})(\delta q_{k/I}^*\hat{q}_{k/I}^*))(\hat{q}_{i/I}\delta q_{i/I}) \tag{122}$$

$$- 2(\delta q_{i/I}^*\hat{q}_{i/I}^*)((\hat{p}_{i/I}\delta q_{i/I})(\delta q_{i/I}^*\hat{q}_{i/I}^*))(\hat{q}_{i/I}\delta q_{i/I}) \tag{123}$$

$$- 2(\delta q_{i/I}^*\hat{q}_{i/I}^*)((\hat{q}_{i/I}\delta p_{i/I})(\delta q_{i/I}^*\hat{q}_{i/I}^*))(\hat{q}_{i/I}\delta q_{i/I}) \tag{124}$$

$$= h_{i r_{ik}}(x_i). \tag{125}$$

Next, we compute the corresponding Jacobian by differentiating $r_{k/i}^i$ with respect to $\delta q_{i/I}$, $\delta q_{k/I}$, $\delta p_{i/I}$ and $\delta p_{k/I}$. First, we can compute

$$H_{r_{k/i}^i, \delta p_{k/I}} = \left.\frac{\partial r_{k/i}^i}{\partial \delta p_{k/I}}\right|_{x_i = \hat{x}_i} \tag{126}$$

$$= \left.2\frac{\partial(\delta q_{i/I}^*\hat{q}_{i/I}^*)((\hat{q}_{k/I}\delta p_{k/I})(\delta q_{k/I}^*\hat{q}_{k/I}^*))(\hat{q}_{i/I}\delta q_{i/I})}{\partial \delta p_{k/I}}\right|_{x_i = \hat{x}_i} \tag{127}$$

$$= 2[|\hat{q}_{i/I}^*|]_R[|\hat{q}_{i/I}|]_L[|\hat{q}_{k/I}^*|]_R[|\hat{q}_{k/I}|]_L \in \mathbb{R}^{4\times 4}, \tag{128}$$

followed by,

$$H_{r_{k/i}^i, \delta q_{k/I}} = \left.\frac{\partial r_{k/i}^i}{\partial \delta q_{k/I}}\right|_{x_i = \hat{x}_i} \tag{129}$$

$$= 2\frac{\partial((\delta q_{i/I}^*\hat{q}_{i/I}^*)((\hat{p}_{k/I}\delta q_{k/I})(\delta q_{k/I}^*\hat{q}_{k/I}^*))(\hat{q}_{i/I}\delta q_{i/I})}{\partial \delta q_{k/I}} + 2\left.\frac{\partial(\delta q_{i/I}^*\hat{q}_{i/I}^*)((\hat{q}_{k/I}\delta p_{k/I})(\delta q_{k/I}^*\hat{q}_{k/I}^*))(\hat{q}_{i/I}\delta q_{i/I}))}{\partial \delta q_{k/I}}\right|_{x_i = \hat{x}_i} \tag{130}$$

$$= 2[|\hat{q}_{i/I}|]_R[|\hat{q}_{i/I}^*|]_L\left([|\hat{q}_{k/I}^*|]_R[|\hat{p}_{k/I}|]_L + [|\hat{p}_{k/I}|]_L[|\hat{q}_{k/I}^*|]_R I^*\right) \in \mathbb{R}^{4\times 4}. \tag{131}$$

Next we have,

$$H_{r_{k/i}^i, \delta q_{i/I}} = \left.\frac{\partial r_{k/i}^i}{\partial \delta q_{i/I}}\right|_{x_i = \hat{x}_i} \tag{132}$$

$$= 2\frac{\partial((\delta q_{i/I}^*\hat{q}_{i/I}^*)((\hat{p}_{k/I}\delta q_{k/I})(\delta q_{k/I}^*\hat{q}_{k/I}^*))(\hat{q}_{i/I}\delta q_{i/I}))}{\partial \delta q_{i/I}} - 2\left.\frac{\partial((\delta q_{i/I}^*\hat{q}_{i/I}^*)((\hat{p}_{i/I}\delta q_{i/I})(\delta q_{i/I}^*\hat{q}_{i/I}^*))(\hat{q}_{i/I}\delta q_{i/I}))}{\partial \delta q_{i/I}}\right|_{x_i = \hat{x}_i} \tag{133}$$

$$= 2[|\hat{q}_{i/I}|]_R[|\hat{p}_{k/I}\hat{q}_{k/I}^*|]_R[|\hat{q}_{i/I}^*|]_R I^* + 2[|\hat{q}_{i/I}^*|]_L[|\hat{p}_{k/I}\hat{q}_{k/I}^*|]_L[|\hat{q}_{i/I}|]_L \tag{134}$$

$$- 2[|\hat{q}_{i/I}|]_R[|\hat{p}_{i/I}\hat{q}_{i/I}^*|]_R[|\hat{q}_{i/I}^*|]_R I^* - 2[|\hat{q}_{i/I}^*|]_L[|\hat{p}_{i/I}\hat{q}_{i/I}^*|]_L[|\hat{q}_{i/I}|]_L \tag{135}$$

$$- 2[|\hat{q}_{i/I}^*|]_L[|\hat{q}_{i/I}|]_R[|\hat{q}_{i/I}^*|]_R[|\hat{p}_{i/I}|]_L - 2[|\hat{q}_{i/I}^*|]_L[|\hat{q}_{i/I}|]_R[|\hat{p}_{i/I}|]_L[|q_{i/I}^*|]_R I^* \in \mathbb{R}^{4\times 4}, \tag{136}$$

and finally,

$$H_{r_{k/i}^i, \delta p_{i/I}} = \left. \frac{\partial r_{k/i}^i}{\partial \delta p_{i/I}} \right|_{x_i = \hat{x}_i} \tag{137}$$

$$= -2 \left. \frac{\partial ((\delta q_{i/I}^* \hat{q}_{i/I}^*)(\hat{q}_{i/I} \delta p_{i/I}))}{\partial \delta p_{i/I}} \right|_{x_i = \hat{x}_i} \tag{138}$$

$$= -2I \in \mathbb{R}^{4 \times 4}. \tag{139}$$

This concludes the computation of the Jacobians of the individual measurement functions in Eq. (52).

B. Full Jacobian for Hard Consensus

In the hard consensus step described in §III.C.2, the measurement Jacobian that satellite i prepares for its neighbour k is given by,

$$\overline{H_{i,k}} = \begin{bmatrix} \overline{H_{i,n_{k,1}}^{n_{k,1}}} & \mathbf{0}_{6\times 6} & \dots & \overline{H_{i,n_{k,1}}^i} & \mathbf{0}_{6\times 6} & \dots & \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} & \dots & \mathbf{I}_{6\times 6} & \mathbf{0}_{6\times 6} & \dots & \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} & \dots & \overline{H_{i,k_{i,k_k}}^i} & \mathbf{0}_{6\times 6} & \dots & \overline{H_{i,k_{i,k_k}}^{k_k}} & \mathbf{0}_{6\times 6} \end{bmatrix} \in \mathbb{R}^{6(k_k+1) \times 12(k_k+1)} \tag{140}$$

$$H_{i,k} = \begin{bmatrix} H_{i,n_{k,1}}^{n_{k,1}} & \mathbf{0}_{8\times 6} & \dots & H_{i,n_{k,1}}^i & \mathbf{0}_{8\times 6} & \dots & \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 6} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 6} & \dots & \mathbf{I}_{8\times 8} & \mathbf{0}_{8\times 6} & \dots & \mathbf{0}_{8\times 6} & \mathbf{0}_{8\times 6} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 6} & \dots & H_{i,k_{i,k_k}}^i & \mathbf{0}_{8\times 6} & \dots & H_{i,k_{i,k_k}}^{k_k} & \mathbf{0}_{8\times 6} \end{bmatrix} \in \mathbb{R}^{8(k_k+1) \times 14(k_k+1)}. \tag{141}$$

References

[1] Johnson, N., "Orbital debris: The growing threat to space operations," *Advances in the Astronautical Sciences*, Vol. 137, 2010, pp. 3–11.

[2] Bandyopadhyay, S., and Quadrelli, M., "Optimal Transport Based Control of Granular Imaging System in Space," *9th International Workshop on Satellite Constellations and Formation Flying*, 2017, pp. 1–11. https://doi.org/10.2514/1.G006001.

[3] Danzmann, K., and the LISA Study Team, "LISA: laser interferometer space antenna for gravitational wave measurements," *Classical and Quantum Gravity*, Vol. 13, No. 11A, 1996, p. A247. https://doi.org/10.2514/6.1995-829.

[4] Amaro-Seoane, P., Andrews, J., Arca Sedda, M., Askar, A., Baghi, Q., Balasov, R., Bartos, I., Bavera, S. S., Bellovary, J., Berry, C. P., et al., "Astrophysics with the laser interferometer space antenna," *Living Reviews in Relativity*, Vol. 26, No. 1, 2023, p. 2. https://doi.org/10.1007/s41114-022-00041-y.

[5] Markley, L., "Attitude Error Representations for Kalman Filtering," *Journal of Guidance Control and Dynamics*, Vol. 26, 2003, pp. 311–317. https://doi.org/10.2514/2.5048.

[6] Farrell, J., "Attitude determination by Kalman filtering," *Automatica*, Vol. 6, No. 3, 1970, pp. 419–430. https://doi.org/10.1016/0005-1098(70)90057-9.

[7] Crassidis, J., Markley, L., and Cheng, Y., "Survey of Nonlinear Attitude Estimation Methods," *Journal of Guidance Control and Dynamics*, Vol. 30, 2007, pp. 12–28. https://doi.org/10.2514/1.22452.

[8] Filipe, N., Kontitsis, M., and Tsiotras, P., "Extended Kalman Filter for spacecraft pose estimation using dual quaternions," *2015 American Control Conference (ACC)*, 2015, pp. 3187–3192. https://doi.org/10.1109/ACC.2015.7171823.

[9] Michieletto, G., and Cenedese, A., "Formation Control for Fully Actuated Systems: a Quaternion-based Bearing Rigidity Approach," *2019 18th European Control Conference (ECC)*, 2019, pp. 107–112. https://doi.org/10.23919/ECC.2019.8795869.

[10] Lee, U., and Mesbahi, M., "Constrained autonomous precision landing via dual quaternions and model predictive control," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 292–308. https://doi.org/10.2514/1.G001879.

[11] Zinage, V., and Bakolas, E., "Koopman operator based modeling and control of rigid body motion represented by dual quaternions," *2022 American Control Conference (ACC)*, 2022, pp. 3997–4002. https://doi.org/10.23919/ACC53348.2022.9867584.

[12] Zu, Y., Sun, C., and Dai, R., "Distributed estimation for spatial rigid motion based on dual quaternions," *53rd IEEE Conference on Decision and Control*, 2014, pp. 334–339. https://doi.org/10.1109/CDC.2014.7039403.

[13] Dong, H., Hu, Q., and Ma, G., "Dual-quaternion based fault-tolerant control for spacecraft formation flying with finite-time convergence," *ISA Transactions*, Vol. 61, 2016, pp. 87–94. https://doi.org/10.1016/j.isatra.2015.12.008.

[14] Filipe, N., and Tsiotras, P., "Adaptive position and attitude-tracking controller for satellite proximity operations using dual quaternions," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, 2015, pp. 566–577. https://doi.org/10.2514/1.G000054.

[15] Yang, J., and Stoll, E., "Adaptive sliding mode control for spacecraft proximity operations based on dual quaternions," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 11, 2019, pp. 2356–2368. https://doi.org/10.2514/1.G004435.

[16] Reynolds, T. P., and Mesbahi, M., "Coupled 6-DOF Control for Distributed Aerospace Systems," *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 5294–5299. https://doi.org/10.1109/CDC.2018.8619618.

[17] Crassidis, J. L., and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, CRC Press, 2004. https://doi.org/10.1201/b11154.

[18] Sabatini, A. M., "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Transactions on Biomedical Engineering*, Vol. 53, No. 7, 2006, pp. 1346–1356. https://doi.org/10.1109/TBME.2006.875664.

[19] Filipe, N., Kontitsis, M., and Tsiotras, P., "Extended Kalman filter for spacecraft pose estimation using dual quaternions," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 9, 2015, pp. 1625–1641. https://doi.org/10.2514/1.G000977.

[20] Alonso, R., Du, J.-Y., Hughes, D., Junkins, J. L., and Crassidis, J. L., "Relative navigation for formation flying of spacecraft," *2001 Flight Mechanics Symposium*, 2001.

[21] Alonso, R., Crassidis, J., and Junkins, J., "Vision-based relative navigation for formation flying of spacecraft," *AIAA guidance, navigation, and control conference and exhibit*, 2000. https://doi.org/10.2514/6.2000-4439.

[22] Sung, K., Peck, C., Majji, M., and Junkins, J. L., "An Optical Navigation System for Autonomous Aerospace Systems," *IEEE Sensors Journal*, Vol. 22, No. 17, 2022, pp. 16862–16873. https://doi.org/10.1109/JSEN.2022.3190910.

[23] Mourikis, A. I., Trawny, N., Roumeliotis, S. I., Johnson, A. E., Ansar, A., and Matthies, L., "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Transactions on Robotics*, Vol. 25, No. 2, 2009, pp. 264–280. https://doi.org/10.1109/TRO.2009.2012342.

[24] Mahmoud, M., and Khalid, H. M., "Distributed Kalman filtering: A bibliographic review," *Control Theory & Applications, IET*, Vol. 7, 2013, pp. 483–501. https://doi.org/10.1049/iet-cta.2012.0732.

[25] Zivan, Y., and Choukroun, D., "Dual quaternion Kalman filters for spacecraft relative navigation," *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 1347. https://doi.org/10.2514/6.2018-1347.

[26] Olfati-Saber, R., "Distributed Kalman Filter with Embedded Consensus Filters," *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 8179–8184. https://doi.org/10.1109/CDC.2005.1583486.

[27] Olfati-Saber, R., and Shamma, J. S., "Consensus Filters for Sensor Networks and Distributed Sensor Fusion," *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 6698–6703. https://doi.org/10.1109/CDC.2005.1583238.

[28] Olfati-Saber, R., "Distributed Kalman filtering for sensor networks," *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 5492–5498. https://doi.org/10.1109/CDC.2007.4434303.

[29] Ryu, K., and Back, J., "Distributed Kalman-filtering: Distributed optimization viewpoint," *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 2640–2645. https://doi.org/10.1109/CDC40024.2019.9029645.

[30] Li, C., Dong, H., Li, J., and Wang, F., "Distributed Kalman filtering for sensor network with balanced topology," *Systems & Control Letters*, Vol. 131, 2019, p. 104500. https://doi.org/https://doi.org/10.1016/j.sysconle.2019.104500.

[31] Roumeliotis, S. I., and Bekey, G. A., "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, 2002, pp. 781–795. https://doi.org/10.1109/TRA.2002.803461.

[32] Binz, J., Hudoba de Badyn, M., Iannelli, A., and Smith, R. S., *Supplementary software for "Distributed Dual Quaternion Extended Kalman Filtering for Spacecraft Pose Estimation"*, 2023 [online]. https://doi.org/10.3929/ethz-b-000510769.

[33] Binz, J., Hudoba de Badyn, M., Iannelli, A., and Smith, R. S., *Supplementary reference for "Distributed Dual Quaternion Extended Kalman Filtering for Spacecraft Pose Estimation"*, 2023 [online]. https://doi.org/10.3929/ethz-b-000619855.

[34] Olfati-Saber, R., "Kalman-Consensus Filter: Optimality, stability, and performance," *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 7036–7042. https://doi.org/10.1109/CDC.2009.5399678.

[35] Gunnam, K. K., Hughes, D. C., Junkins, J. L., and Kehtarnavaz, N., "A vision-based DSP embedded navigation sensor," *IEEE Sensors Journal*, Vol. 2, No. 5, 2002, pp. 428–442. https://doi.org/10.1109/JSEN.2002.806212.

[36] Kreiss, S., Bertoni, L., and Alahi, A., "OpenPifPaf: Composite fields for semantic keypoint detection and spatio-temporal association," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 8, 2021, pp. 13498–13511. https://doi.org/10.1109/TITS.2021.3124981.

[37] Kreiss, S., Bertoni, L., and Alahi, A., "PifPaf: Composite Fields for Human Pose Estimation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11977–11986. https://doi.org/10.1109/CVPR.2019.01225.

[38] Markley, L., Cheng, Y., Crassidis, J., and Oshman, Y., "Averaging Quaternions," *Journal of Guidance, Control, and Dynamics*, Vol. 30, 2007, pp. 1193–1196. https://doi.org/10.2514/1.28949.

[39] Gonzalez, A., "Measurement of Areas on a Sphere Using Fibonacci and Latitude Longitude Lattices," *Mathematical Geosciences*, Vol. 42, 2010, pp. 49–64. https://doi.org/10.1007/s11004-009-9257-x.

[40] Binz, J., Hudoba de Badyn, M., Iannelli, A., and Smith, R. S., *Supplementary video for "Distributed Dual Quaternion Extended Kalman Filtering for Spacecraft Pose Estimation"*, 2023 [online]. https://doi.org/10.3929/ethz-b-000606978.