Distributed Motion Estimation of Space Objects Using Dual Quaternions

Yue Zu*, Unsik Lee†, and Ran Dai‡

This paper examines the motion estimation problem for space objects using multiple image sensors in a connected network. The objective is to increase the estimation precision of relative translational and rotational motions based on integrated dual quaternion representations and cooperation between connected sensors. The relative motion of space objects is first formulated using dual elements to express its kinematics and dynamics. Two modular optimization approaches, namely dual decomposition and distributed Newton methods, for decomposing this cooperative estimation problem among the sensors is then proposed. Simulation results from single sensor estimation and two distributed estimation frameworks are provided and compared.

I. Introduction

A number of space missions depend on accurate estimation of the relative motion of space objects. Examples include rendezvous, formation flight, and debris avoidance. However, for space objects flying with high translational and rotational speed, it is challenging to obtain precise information of their relative position and orientation.

One of the challenges stems from the coupled translational and rotational motion of the space object, which requires a compact and efficient mathematical model to describe its combined dynamics. A traditional expression describing the spatial rotation is the rotation matrix represented by Euler angles. However, the terms in the rotation matrices are nonlinear due to the trigonometric functions. Moreover, matrices in terms of Euler angles may cause singularities. Quaternions have been introduced as an alternative mathematical tool for calculating three dimensional (3D) rotations. Quaternions are particularly useful because they avoid singularities and reduce the expensive computational load created by Euler angle expressions [1]. Quaternions have also proven to be an effective representation of the orientation of a rigid body. Furthermore, quaternions are playing an indispensable role in dynamical systems due to their unambiguous expression of spatial rotations [2]. However, quaternions cannot represent rotations and translations simultaneously. The dual quaternion was proposed by W. K. Clifford to address this concern [3]. The representation has since been recognized as a non-singular, efficient and compact tool applicable to many fields.

The Extended Kalman Filter (EKF) algorithm is a powerful technique for parameter estimation. Its advantage over a standard Kalman Filter is its allowance of nonlinearity in the system. Current work regarding spatial motion estimation mainly focuses on using EKF with observations from a signal sensor [4-7]. If traditional image based sensors are used for measurement, part of the measurement data will be not available when the space object flies out of the sensors’ field of view or when their vision is blocked by interference. The missing information of the observed object due to such visual constraints or malfunctions adds complexity to the real-time estimation of the relative motion. Furthermore, the complicated motion, huge amount of observation data, and subsequent computational burden motivate the investigation of an efficient estimation algorithm to process the observed data in real time.

Distributed estimation technology has been commonly used in process control, signal processing, and information systems [8-10]. A subset of these efforts have generally been focused on the integration of

*Research Assistant, Aerospace Engineering Department, Iowa State University, Ames, IA, 50011-2271, Email: yzu@iastate.edu
†Research Associate, Department of Aeronautics and Astronautics, University of Washington Seattle, Box 352400, WA, 98195-2400. Email: unsik@u.washington.edu
‡Assistant Professor, Aerospace Engineering Department, Iowa State University, Ames, IA, 50011-2271, Email: dairan@iastate.edu
measurements from all the sensors into a common estimate without a centralized processor. For example, the work reported in [11] has successfully applied the data fusion strategy in distributed orbit determination where the individual sensor in a connected network will process its estimates before communicating to the other nodes. Even though each sensor can only provide part of the measurements due to the field-of-view constraint during the observation interval, the distributed estimation algorithm can produce accurate estimates of orbital parameters and prediction of subsequent motion via communication and coordination. The estimation task will be more complicated when both translational and rotational motions are considered. Considering the challenging observation environment in space and the constraints of individual sensors, in this work we propose a distributed estimation scheme via the dual decomposition methodology and subgradient method [12] to estimate the space object’s motion using multiple sensors, where the kinematics of the object is based on the dual quaternion model.

The dual decomposition approach has been applied not only in optimization but also in estimation problems [13, 14]. In the work [14], the authors have implemented dual techniques combined with the subgradient method in distributed estimation where they use multiple sensors to obtain the observation data. Motivated by the advantages of these works, this paper applies the dual decomposition technique in a distributed EKF algorithm to estimate spatial motion of space objects. By solving each subproblem individually and coordinating the estimation consensus constraints through the subgradient method, we can find the global optimal solution. Inspired by the rapid convergence of the Newton’s method in solving network utility maximization problems [15, 16], we also propose to design a Newton-type distributed estimation algorithm aimed at increasing the rate of convergence.

In the remainder of the paper, we first introduce the definition of the dual quaternion and describe kinematics and dynamics representations based on dual quaternions in Section II. In Section III we proceed to present the EKF algorithm by developing the state transition equation and measurement equation. Section IV describes the dual decomposition and subgradient method followed by the application of the EKF algorithm. Next we discuss a simulation example and its comparison to a single EKF in Section V followed by concluding remarks.

II. Dual Quaternion

A. Definition of Dual Quaternion

Suppose that the relationship between the inertially fixed frame \( O \) and the body frame \( B \) is as shown in Fig. 1. At each instance, the configuration space for position and orientation of the rigid body is described by a \( 4 \times 4 \) homogeneous transformation matrix. \( \text{SE}(3) \) is the set of all rigid body transformations in three dimensional space:

\[
\text{SE}(3) = \left\{ T \in \mathbb{R}^{4 \times 4} \mid T = \begin{bmatrix} R & t_o \\ 0 & 1 \end{bmatrix}, R \in \text{SO}(3), t_o \in \mathbb{R}^3 \right\},
\]

where \( t_o \) denotes the position vector to the body frame \( B \) with inertial frame components and \( R \) is a rotation matrix.

As unit quaternions parameterize \( \text{SO}(3) \), a unit dual quaternion, introduced by Clifford [17], can be used to define a rigid body rotation \( (q \in \mathbb{S}^3 \times \mathbb{R}^3 \mapsto \text{SE}(3)) \). The geometric difference between the frame \( B \) with respect to the inertially fixed frame \( O \) can be expressed by a translation \( t_o \) (represented in frame \( O \)) followed by a rotation \( q_r \), which is represented in the form of \textit{unit dual quaternions} as

\[
\tilde{q} = q_r + \varepsilon \left( \frac{1}{2} t_o \otimes q_r \right),
\]

where \( t_o \) represents the translation vector \( t \) with respect to the frames \( O \) expressed in frame \( O \) and a zero scalar part, \( \tilde{q} = q_r + q_d \) are real and dual part, respectively. Converting between \( t_o \) and \( t_b \) is governed by quaternion rotation operator as follows,

\[
t_o = q_r \otimes t_b \otimes q_r^*, \quad t_b = q_r^* \otimes t_o \otimes q_r,
\]

where \( q_r^* \) is the conjugate of \( q_r \).
B. Dual Quaternion Kinematics and Dynamics

The kinematics of a rigid body can be represented in terms of dual quaternion as [18]

\[ \dot{q}_r = \frac{1}{2} \omega_o \otimes q_r = \frac{1}{2} q_r \otimes \omega_o \]  

(2.5)

or

\[ \dot{q}_d = \frac{1}{2} (t_o \otimes q_r)' = \frac{1}{2} t_o \otimes q_r + \frac{1}{2} t_o \otimes q_r = \frac{1}{2} t_o \otimes q_r + \frac{1}{2} t_o \otimes \omega_o \otimes q_r \]  

(2.6)

where \( \omega_o \) and \( \omega_b \) are vector quaternions formed from the angular velocity vector and a zero scalar part, expressed in original frame and body frame. Summarizing (2.5) and (2.6), we get the dual quaternion kinematic equation in the form of

\[ \dot{\tilde{q}} = \frac{1}{2} \tilde{q} \hat{\omega}_b, \]  

(2.7)

where \( \tilde{\omega}_b = \omega_b + \epsilon \nu_b = \omega_b + \epsilon (\dot{t}_b + \omega_b \times t_b) \) is called the twist [19]. The translational and rotational motions of a fully actuated rigid body are expressed by the rate of change of linear and angular momentum through the following differential equations [20],

\[ F = \left[ \frac{d}{dt} (m \nu) \right]_B = m \dot{\nu} + \omega_b \times m \nu, \]  

(2.8)

\[ T = \left[ \frac{d}{dt} (J \omega) \right]_B = J \dot{\omega}, \]  

(2.9)

where \( [d(\cdot)/dt]_B \) denotes the time derivative represented in the body frame \( B \), \( m \in \mathbb{R} \) denotes the mass of the rigid body, \( J \in \mathbb{R}^{3 \times 3} = \text{diag}(J_1, J_2, J_3) \) denotes its inertia matrix along the body frame, and \( \omega_b, \nu_b \in \mathbb{R}^3 \) denote, respectively, the angular and translational velocities of the rigid body represented in the body frame. Moreover, \( F \) and \( T \) denote the external translational force and torque acting on the rigid body, expressed in the body frame. Note that in order to focus on the feasibility of the dual quaternion-based algorithm, we assume that uncertainties and all external disturbances on the rigid body are negligible. Researchers have implemented the extended Kalman filter to estimate the translational and rotational motion of the concerned object using a single image sensor [18, 21].

III. EKF Algorithm Based on Dual Quaternion Representations

A. Extended Kalman Filter Algorithm

For a discrete and linear system, the state transition and observation functions are defined as

\[ x_{k+1} = F x_k + w_k, \]  

(3.10)

\[ z_{k+1} = H x_k + v_k, \]  

(3.11)
Protocol 1 Extended Kalman Filtering Algorithm

\textit{Initialization:} \( P_0^- = E[(x_0 - x_0^-)(x_0 - x_0^-)^T] \), \( x_0^- \)

\textbf{for} \( k \gets 1 \) \textbf{to} number of sample data \( k_{\text{max}} \) \textbf{do}

- Update linearized observation matrix: \( H_k = \frac{\partial h(x)}{\partial x} \mid_{x=x_k^-} \)
- Compute Kalman gain: \( K = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1} \)
- Compute the corrected estimates \( x_k^- = x_k^- + K(z_k - H_k x_k^-) \)
- Update linearized state matrix: \( F_k = \frac{\partial f(x)}{\partial x} \mid_{x=x_k^-} \)
- Compute 1-step prediction and associated error covariance at step \( k+1 \): \( x_{k+1}^- = F_k x_k^- \), \( P_{k+1}^- = F_k P_k^- F_k^T + Q \)
- Go to Step 3 and increment \( k \) by one

\textbf{end}

where \( x \) is \( n \times 1 \) state variable, \( z \) is the \( m \times 1 \) measurement vector, \( w \) is \( n \times 1 \) system noise which is zero mean Gaussian with covariance matrix \( Q \in \mathbb{R}^{n \times n} \), and \( v \) is \( m \times 1 \) measurement noise which is Gaussian with covariance matrix \( R \in \mathbb{R}^{m \times m} \), \( F \) and \( H \) are the state and measurement matrices, respectively. When considering a nonlinear system with a nonlinear observation model, \( F \) and \( H \) are time varying matrices. Equations (3.10) and (3.11) are expressed as follows after linearization,

\[
\begin{align*}
  x_{k+1}^- &= F_k x_k^- + w_k, \\
  z_{k+1}^- &= H_k x_k^- + v_k,
\end{align*}
\]

(3.12) (3.13)

where \( F_k \) and \( H_k \) are state transition matrix and observation matrix at time interval \( k \). The EKF algorithm has been proved as a Minimum Mean Square Error (MMSE) estimator. We omit the proof and give the recursive algorithm below to find the estimates:

B. State Transition Model

The state transition model derived from dual kinematics indicates the evolution from current to future states. First we select the following states as the representation of the spatial motion of a space object,

\[
x = [q_r^T \ q_d^T \ \omega_b^T \ v_b^T]^T.
\]

By discretizing Eqs.(2.7)-(2.9) using sample time \( \tau \), the discrete spatial motion of a space object is represented by

\[
\begin{align*}
  q_r(k+1) &= q_r(k) + \frac{\tau}{2} q_r(k) \otimes \omega_b(k) \\
  q_d(k+1) &= q_d(k) + \frac{\tau}{2} q_d(k) \otimes v_b(k) + \frac{\tau}{2} q_d(k) \otimes \omega_b(k) \\
  v_b(k+1) &= v_b(k) + E \frac{\tau}{m} - \frac{\tau}{m} \omega_b(k) \otimes v_b(k) \\
  \omega_b(k+1) &= \omega_b(k) + \tau J^{-1} T - \tau J^{-1} (\omega_b(k) \times J \omega_b(k)).
\end{align*}
\]

(3.14)

By linearization of the above equation, the state transition matrix \( F(x_k) = \frac{\partial f(x)}{\partial x} \mid_{x=x_{k-1}} \in \mathbb{R}^{n \times n} \) at time point \( k \) is expressed as

\[
F(x_k) = \begin{bmatrix}
  I + \frac{\tau}{2} \Omega_b & 0 & \frac{\tau}{2} Q_r & 0 \\
  \frac{\tau}{2} V_b & \frac{\tau}{2} \Omega_b + I & \frac{\tau}{2} Q_d & \frac{\tau}{2} Q_r \\
  0 & 0 & M + I & 0 \\
  0 & 0 & N & S + I
\end{bmatrix}_k,
\]

(3.15)

where \( \Omega_b = \begin{bmatrix} 0 & -\omega_b^T \\
    \omega_b & -K(\omega_b) \end{bmatrix} \), \( Q_r = \begin{bmatrix} q_{r0} & -q_{r1} \\
    q_{r1} & q_{r0} + K(q_{r1}) \end{bmatrix} \), \( V_b = \begin{bmatrix} 0 & -\vec{v}_b^T \\
    \vec{v}_b & -K(\vec{v}_b) \end{bmatrix} \), \( Q_d = \begin{bmatrix} q_{d0} & -q_{d1} \\
    q_{d1} & q_{d0} + K(q_{d1}) \end{bmatrix} \), the skew-symmetric matrix \( K(\vec{y}) = \begin{bmatrix} 0 & -y_3 & y_2 \\
    y_3 & 0 & -y_1 \\
    -y_2 & y_1 & 0 \end{bmatrix} \), \( M = -\tau \)

\[
\begin{bmatrix}
  0 & 0 & 0 & 0 \\
  0 & 0 & \omega_z(J_3 - J_2)/J_1 & \omega_y(J_3 - J_2)/J_1 \\
  0 & \omega_z(J_1 - J_3)/J_2 & 0 & \omega_x(J_1 - J_3)/J_2 \\
  0 & \omega_z(J_2 - J_1)/J_3 & \omega_x(J_2 - J_1)/J_3 & 0
\end{bmatrix}.
\]
\[
N = -\tau \begin{bmatrix}
0 & 0_{1 \times 3} \\
0_{3 \times 1} & -K(\vec{\omega}_b)
\end{bmatrix}, \quad \text{and} \quad S = -\tau \begin{bmatrix}
0 & 0_{1 \times 3} \\
0_{3 \times 1} & -K(\vec{\omega}_b)
\end{bmatrix}.
\]

C. Observation Model

Measurement data in this model is a set of images that project 3D trajectories to 2D images using high speed cameras. Several feature points on the surface of the concerned object are identified first. Our purpose is to find out the relationship between the coordinates in the body frame and the coordinates in the camera/sensor frame. Similar to Eq. (2.3), we rewrite dual quaternion transformation as

\[
p_o = \hat{q} \hat{p}_b \hat{q}^* ,
\]

where \( p_o = 1 + (x_o i + y_o j + z_o k) \varepsilon \) and \( p_b = 1 + (x_b i + y_b j + z_b k) \varepsilon \) denote the dual representation of the featured point in camera/sensor frame and body frame, respectively; in Eq.(3.16) \( \hat{q}^* \) denotes the dual quaternion conjugate. The notation \( \hat{q} \hat{p} \) refers to dual quaternion multiplication. By substituting \( p_o \) and \( p_b \) using the definition of dual quaternions, Eq.(3.16) is written as

\[
\begin{bmatrix}
0 \\
x_o \\
y_o \\
x_o
\end{bmatrix} = Q_r \bar{Q}^*_r Q_d^* + Q_d q_r^* + Q_d q_r^* ,
\]

where \( \bar{Q}_r = \begin{bmatrix}
q_{10} & -\vec{q}^*_r \\
\vec{q}_r & q_{10} - K(\vec{q}_r)
\end{bmatrix} \). The transformation can be divided into two separate steps including the first term (rotation) and the last two terms (translation) in (3.17). So far, the coordinate transformation of feature point from body frame to camera/sensor frame has been established. The corresponding projection of the feature point on the image plane is determined by [22]

\[
x_i = F_c x_o \\
y_i = F_c y_o .
\]

In this paper, three feature points are identified on the space object surface. Therefore, the measurements are composed of three pairs of coordinates expressed as \( h(x) = [x_1^i, y_1^i, x_2^i, y_2^i, x_3^i, y_3^i]^T \). In addition, the unit constraint on the real part and the orthogonal constraint are considered as additional measurements. Hence, \( q_r^T q_r = 1 \) and \( q_r^T q_d = 0 \) are added to the measurements \( h(x) \). Under these settings, the measurement matrix is derived as

\[
H_k = \left. \frac{\partial h(x)}{\partial \mathbf{x}} \right|_{x=x_k^-} .
\]

IV. Distributed Estimation

A. Motivation

The motion estimation for a space object by multiple sensors can be formulated as minimizing the estimation error function

\[
J = \min \sum_{i=1}^{N} f^i(\mathbf{x})
\]

with estimates agreement

\[
\hat{x}_i = \hat{x}_j, \quad \forall i \in N, i \neq j,
\]

where \( \mathbf{x} \in \mathbb{R}^n \) represents the estimation state, \( i \) is the index of the sensor, and \( N \) is the number of image sensors. The measurement obtained from individual sensor, \( p_i, i = 1, \ldots, N \), is the projection of the feature
point \( p_0 \) on the corresponding image frame, as demonstrated in Fig.2. Due to the visual constraints of the single sensor, we may not be able to retrieve the complete space object motion from one sensor. By exchanging information with neighboring sensors in a connected network, we aim at estimating both translational and rotational motion of the space object with the overall minimum estimation error.

**Figure 2. Distributed estimation of space object motion by multiple image sensors**

**B. Dual Decomposition and Subgradient Method**

For a nonlinear optimization problem of minimizing the objective function \( f(x) \) under equality constraints \( h_i(x) = 0 (i = 1, \ldots, p) \) and inequality constraints \( g_i(x) \leq 0 (i = 1, \ldots, m) \), its Lagrangian has the form:

\[
L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{p} \mu_i h_i(x) + \sum_{i=1}^{m} \lambda_i g_i(x).
\] (4.22)

If the objective and constraint functions can be expressed in the summation form as

\[
f(x) = \sum_{k=1}^{K} f^k(x^k), \quad g_i(x) = \sum_{i=1}^{K} g_i^k(x^k), \quad h_i(x) = \sum_{i=1}^{K} h_i^k(x^k),
\] (4.23)

by partitioning the state vector \( x \) into subvectors \( x = (x^1, \ldots, x^K) \), the Lagrangian can be written as

\[
L(x, \lambda, \mu) = \sum_{k=1}^{K} L_k(x, \lambda, \mu) = \sum_{k=1}^{K} (f^k(x^k) + \sum_{i=1}^{p} \mu_i h_i^k(x^k) + \sum_{i=1}^{m} \lambda_i g_i^k(x^k)).
\] (4.24)

The above function can be decomposed into \( K \) subproblems according to the subvector \( x^k \), where \( k = 1, \ldots, K \). In this case, each subproblem can be solved by minimizing the dual function

\[
L_k^D(\lambda, \mu) = \min_{x^k} (f^k(x^k) + \sum_{i=1}^{p} \mu_i h_i^k(x^k) + \sum_{i=1}^{m} \lambda_i g_i^k(x^k))
\] (4.25)

for a given pair of multipliers \( (\lambda^k, \mu^k) \). The dual function defined as

\[
L_k^D(\lambda, \mu) = \inf_{x} L(x, \lambda, \mu)
\] (4.26)

is always concave, thus the dual problem of the summation of the subproblems

\[
\max_{(\lambda, \mu)} L_K^D(\lambda, \mu)
\] (4.27)
is a convex optimization problem and can be solved by the subgradient method.

The subgradient method is an iterative procedure based on the ascent direction for the dual problem. At each sequence \( j \), assuming the multipliers \((\lambda_j, \mu_j)\) are given, the subgradient at this point is expressed as

\[
d_j = \begin{pmatrix} g(x_j) \\ h(x_j) \end{pmatrix}.
\]

Then the multipliers are updated as follows:

\[
\begin{align*}
\lambda_{j+1} &= \max(0, \lambda_j + \alpha_j g(x_j)) \\
\mu_{j+1} &= \mu_j + \alpha_j h(x_j),
\end{align*}
\]

where \( \alpha_j \) is the step size that will control the convergence speed of the subgradient method. The maximum iteration generally chosen based on the stopping criteria.

C. Dual decomposition in Distributed EKF

In a connected sensor network as shown in Fig. 3, we assume that each sensor can communicate with its neighbors. There is no central processor and the network is not fully connected. However, as long as there is a connection which is defined by the entries of the adjacency matrix \( A(G) \) between any two nodes in the network, the two connected sensors can communicate with each other. Furthermore, they are able to spread the information among the connected network. In such a system, the information filter or some other data fusion algorithm that requires a fully connected network cannot be applied to such a partially connected system. In addition, the fully connected network requires large data communication and storage which may make its implementation difficult. Furthermore, without consensus constraints, there is no limitation to converge the final result to the average consensus.

![Figure 3. Communication and framework of multi-sensor network](image)

For a network with \( N \) sensors depicted in Fig. 3, the index of neighbors for sensor \( i \) is determined by the entries \( A(G) \), \( i = 1, \ldots, N, i \neq j \). If \( A_{i,j} = 1 \), then sensor \( i \) can communicate with sensor \( j \) and we will expect the estimates obtained from sensor \( i \), \( \hat{x}(i) \), to be identical to \( \hat{x}(j) \) as well as the other estimates from other connected neighbors. By passing the identity request from one node to the other in the network, we can transfer the consensus request in the system. With the neighborhood consensus constraints on the estimates, we have the following relationships:

\[
a_{i,j} \hat{x}(i) - a_{i,j} \hat{x}(j) = 0, \quad i = 1, \ldots, N, \quad i > j,
\]

where \( a_{i,j} \) denotes the element \( A_{i,j} \) in matrix \( A \). If there is a connection between sensors \( i \) and \( j \), the consensus condition expressed in Eq. (4.30) will ensure \( \hat{x}(i) = \hat{x}(j) \). Since matrix \( A \) is symmetric, we will have \( a_{i,j} = a_{j,i} \).

The Kalman filter algorithm for the system described in (3.10) and (3.11) finds the maximum-likelihood estimates \( \hat{x}_k \) to minimize the least-square objective function:

\[
J = (\hat{x}_k - x_k^-)^T (P_k^-)^{-1} (\hat{x}_k - x_k^-) + (z_k - H_k x_k^-)^T R_k^{-1} (z_k - H_k x_k^-)
\]

(4.31)
As we mentioned in Eq. (4.21), Lagrange multipliers $\lambda_i$ are introduced for the estimation consensus constraints in EKF objective function such that

$$L = \sum_{i=1}^{N} [(\hat{x}^i_k - x^i_k)^T (P^{-1}_k)^i (\hat{x}^i_k - x^i_k) + (z^i_k - H^i_k x^i_k)^T R^{-1}_k (z^i_k - H^i_k x^i_k)] + \sum_{i=1}^{N} \sum_{h=i+1}^{N} \lambda_i,h [a_{i,h} \hat{x}^i_k - a_{i,h} \hat{x}^h_k],$$

where $L$ is the Lagrangian, $\hat{x}^i_k$ denotes the maximum-likelihood estimate obtained from the observation of the $i$th sensor at time instant $k$, and as $L = \sum_{i=1}^{N} L_i$, where $x^{i,-}_k$ represents the corresponding 1-step prediction. Obviously, the Lagrangian is decomposable

$$L_i = (\hat{x}^i_k - x^{i,-}_k)^T (P^{-1}_k)^i (\hat{x}^i_k - x^{i,-}_k) + (z^i_k - H^i_k x^{i,-}_k)^T R^{-1}_k (z^i_k - H^i_k x^{i,-}_k) + \sum_{h=i+1}^{N} \lambda_i,h [a_{i,h} \hat{x}^i_k - a_{i,h} \hat{x}^h_k], \quad i = 1, \ldots, N. \quad (4.32)$$

These subproblems are independent and can be solved individually. By substituting $z^i_k = H^i_k \hat{x}^i_k$ into Eq. (4.32), we take the partial derivative of (4.32) to determine $\hat{x}^i_k$:

$$\frac{dL_i}{d\hat{x}^i_k} = (P^{-1}_k)^i (\hat{x}^i_k - x^{i,-}_k) - (H^i_k)^T R^{-1}_k (z^i_k - H^i_k x^{i,-}_k) - \sum_{h=i+1}^{N} \lambda_i,h a_{i,h} + \sum_{h=1}^{i-1} \lambda_{i,h} a_{h,i}, \quad i = 1, \ldots, N. \quad (4.33)$$

The Lagrangian achieves its minimum if Eq. (4.33) is zero. Hence, we get the updated function of $\hat{x}^i_k$ at time instant $k$:

$$\hat{x}^i_k = x^{i,-}_k + P_k (H^i_k)^T R^{-1}_k (z^i_k - H^i_k x^{i,-}_k) + P_k (- \sum_{h=i+1}^{N} \lambda_{i,h} a_{i,h} + \sum_{h=1}^{i-1} \lambda_{h,i} a_{h,i}), \quad i = 1, \ldots, N. \quad (4.34)$$

In the above equation $P_k (H^i_k)^T R^{-1}_k = K$ is the Kalman gain which is similar to the Kalman gain update in EKF algorithm. The only distinction is the last additional term. The extra distinction can be handled as an adjustment of estimates to satisfy consensus constraints. By using the subgradient method, Lagrange multipliers are updated as

$$\lambda^{i+1}_{i,h} = \lambda^i_{i,h} + \alpha_j a_{i,h} (\hat{x}^i_k - \hat{x}^h_k), \quad i = 1, \ldots, N, \quad h = i + 1, \ldots, N, \quad (4.35)$$

where $\alpha_j$ is the step size.

### D. Distributed Newton Method

Intuitively, multi-sensor networks are expected to improve the estimation precision of spatial rigid motion using compact dual quaternion representations. However, the dual-decomposition-based distributed estimation approach requires iterative coordination between networked sensors. For each iteration, the individual sensor must update state variables using EKF and the new estimates are figured out later. It is imperative to develop a faster convergent distributed estimation algorithm that will lead to minimum estimation error with less iterative coordination. Inspired by the rapid convergence of the Newton’s method in solving network utility maximization problems, we propose a Newton-type distributed estimation algorithm, aiming at increasing the rate of convergence.

More generally, Eq. (4.20) and (4.21) can be extended to the following general constrained optimization problem

$$J = \min_{x^i, \ldots, x^n} \sum_{i=1}^{n} f^i(x^i)$$

s.t. $C x = b, \quad (4.36)$

where $x^i \in \mathbb{R}^{n^i}, i = 1, \ldots, n$, $C \in \mathbb{R}^{m^i \times n^i}$, and $b \in \mathbb{R}^{m^i}$. From a feasible starting point $x(0)$, the iterative approach for solving the above optimization is expressed as

$$x(j + 1) = x(j) + s(j) \Delta x(j), \quad (4.37)$$
where $\Delta x(j)$ and $s(j)$ are the Newton direction and step size, respectively, at iteration step $j$. The Newton direction $\Delta x(j)$ is obtained by solving the following linear function,

$$
\begin{pmatrix}
\nabla^2 f[x(j)] & C^T \\
C & 0
\end{pmatrix}
\begin{pmatrix}
\Delta x(j) \\
w(j)
\end{pmatrix}
= -
\begin{pmatrix}
\nabla f[x(j)] \\
0
\end{pmatrix},
$$

(4.38)

where $\nabla^2 f[x(j)]$ and $\nabla f[x(j)]$ are the Hessian matrix and the gradient of the objective function evaluated at $x(j)$, respectively, and $w(j)$ is the dual variable of the linear constraint. For notation simplicity, we use $\nabla^2 f_j = \nabla^2 f[x(j)]$ and $\nabla f_j = \nabla f[x(j)]$ in the following text. From (4.38), we get

$$
\Delta x(j) = - (\nabla^2 f_j)^{-1}(\nabla f_j + C^T w(j))
$$

(4.39)

$$(C(\nabla^2 f_j)^{-1}C^T)w(j) = -C(\nabla^2 f_j)^{-1}\nabla f_j.
$$

(4.40)

Since the objective function is decomposable in terms of $x^j$, the diagonal elements in Hessian matrix can be calculated individually, denoted as $\nabla^2 f_j(x^i) = \partial^2 f/\partial x^i(j)^2$. However, finding $w(j)$ requires global information to calculate the matrix $(C(\nabla^2 f_j)^{-1}C^T)^{-1}$. Work in [16] has introduced a distributed inexact Newton method to compute the $w(j)$ using an iterative method. We apply it here to generate the estimation algorithm in a distributed manner.

The estimation problem with objective of (4.31) and consensus constraint of (4.30) can be handled as one of the general constrained optimization problems described in (4.36). We use $\nabla f_j(x^i_k)$ and $\nabla^2 f_j(x^i_k)$ to represent the elements in the gradient vector and Hessian corresponding to $x^i$ at iteration $j$, respectively. Their expressions in solving estimation problem of (4.31) with constraint (4.30) are found as

$$
\nabla f_j(x^i_k) = (P_k^{l})^{-1}(x^i_k(j) - x^j_k(j)) - (H_k^i)T R^{-1}(z^i_k - H_k^i x^i_k(j))
$$

(4.41)

$$
\nabla^2 f_j(x^i_k) = (P_k^{l})^{-1} + (H_k^i)T R^{-1}H_k^i.
$$

(4.42)

By substituting $\nabla f_j(x^i_k)$ and $\nabla^2 f_j(x^i_k)$ in (4.40), the Newton direction for updating $x^i$ only is determined by,

$$
\Delta x^i_k(j) = -[\nabla^2 f_j(x^i_k)]^{-1}\nabla f_j(x^i_k) - [(\nabla^2 f_j(x^i_k))^{-1}C^T w(j)].
$$

The first term in the above equation is dependent on states and measurements of sensor $i$ only. Furthermore, $w(j)$ can be solved by (4.40) through the decentralized method in [16]. Therefore, we can find the estimates in a distributed manner. The major steps of obtaining $w(j)$ is described below and more details can be referred to [16]. We firstly define two matrices,

$$
D(j)u = (C(\nabla^2 f_j)^{-1}C^T)u
$$

(4.43)

$$
E(j) = C(\nabla^2 f_j)^{-1}C^T - D(j).
$$

(4.44)

Let $w_0$ be an arbitrary initial vector and the sequence $w_s$ is generated by

$$
w_{s+1} = D(j)^{-1}(-C(\nabla^2 f_j)^{-1}\nabla f_j) - D(j)^{-1}E(j)w_s,
$$

(4.45)

When $s \to \infty$, the limit $w_s = w(j)$ is the solution to Eq. (4.40).

V. Simulation Example

A. Initial Location of Cameras/Sensors

In the simulation, two sensors/cameras are assumed to be implemented in a spacecraft for observing the relative spatial motion of a near earth asteroid. The sensors’ initial positions are illustrated in Fig. 4. Fig. 4 also shows the asteroid initial body frame $x_0 - y_0 - z_0$ and the initial position of feature points in the body frame. Specifically, camera 1 is below $x_b - y_b - z_b$ with 10 meters shifting along $-z_b$. And camera 2 is at the left hand side of frame $x_b - y_b - z_b$ with 10 meters shifting along $-y_b$ and a $\pi$ rotation with respect to $-x_b$. The spacecraft is in a circular orbit around the earth. Table.1 lists the corresponding parameters used in this simulation.
Table 1. Parameter Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude of the spacecraft</td>
<td>590 km</td>
</tr>
<tr>
<td>Radius of the earth</td>
<td>6378.1370 km</td>
</tr>
<tr>
<td>Pixel spacing along $x$, $P_x$</td>
<td>$2 \times 10^{-5}$ m/pixel</td>
</tr>
<tr>
<td>Pixel spacing along $y$, $P_y$</td>
<td>$2 \times 10^{-5}$ m/pixel</td>
</tr>
<tr>
<td>Sample period, $T$</td>
<td>0.02 s</td>
</tr>
<tr>
<td>Focal length, $F_c$</td>
<td>0.017 m</td>
</tr>
<tr>
<td>Measurement noise variance(sensor 1)</td>
<td>0.01 pixel$^2$</td>
</tr>
<tr>
<td>Measurement noise variance(sensor 2)</td>
<td>0.06 pixel$^2$</td>
</tr>
<tr>
<td>Object(asteroid) translation velocity $v_b$</td>
<td>1 km/s (along $x_b$ axis)</td>
</tr>
<tr>
<td>Object(asteroid) angular velocity $\omega_b$</td>
<td>$2\pi$ rad/s (direction $+x_b$)</td>
</tr>
</tbody>
</table>

B. Results

Two simulation results are given below. The first one is carried out by two individual cameras without cooperation. In the second simulation, distributed EKF is implemented based on cooperative estimation between the two sensors. Figures 5-7 demonstrate the estimates of the real part $q_r$, the relative position, and angular velocity, from individual sensors. Since camera 1 is below the object and camera 2 is at the right hand side of it, the observability of camera 1 is lower than camera 2 along $z$ axis, which leads to poor estimates of $z$ from camera 1 in Fig. 6.

Figures 8 and 9 provide the corresponding results using cooperative sensors based on the distributed dual decomposition and Newton methods, respectively. To save space, only angular velocity estimates are provided here. We note that both distributed methods will converge to the estimation consensus. However, estimate from the distributed Newton method converge to the consensus much faster than that from the dual decomposition method. Furthermore, by comparing the MSE of estimated angular velocity terms calculated from three methods in Table 2, we found the poor observation data obtained from sensor 2 affects the estimation precision of Sensor 1 in the dual decomposition method, which is avoided in Newton’s method. By comparison, we reach the conclusion that the distributed Newton method has faster convergence rate and higher precision than the dual decomposition method.
Figure 5. Comparison of real and estimated real part of dual quaternion from individual sensors

Figure 6. Comparison of real and estimated relative position from individual sensors

Figure 7. Comparison of real and estimated angular velocities from individual sensors
Figure 8. comparison of dual decomposition and Newton methods for estimates of angular velocities

Figure 9. Comparison of Newton method and individual sensors for estimates of angular velocity $\omega_x$

Table 2. MSE of angular velocity terms calculated from single EKF, dual decomposition and Newton method

<table>
<thead>
<tr>
<th>$\omega$ from Sensor 1</th>
<th>EKF</th>
<th>Distributed EKF</th>
<th>Newton method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega_x$</td>
<td>0.5233</td>
<td>1.0854</td>
<td>0.4620</td>
</tr>
<tr>
<td>$\omega_y$</td>
<td>0.0606</td>
<td>0.0295</td>
<td>0.0185</td>
</tr>
<tr>
<td>$\omega_z$</td>
<td>0.0373</td>
<td>0.0179</td>
<td>0.0166</td>
</tr>
<tr>
<td>$\omega$ from Sensor 2</td>
<td>EKF</td>
<td>Distributed EKF</td>
<td>Newton method</td>
</tr>
<tr>
<td>$\omega_x$</td>
<td>1.1672</td>
<td>1.0473</td>
<td>0.4620</td>
</tr>
<tr>
<td>$\omega_y$</td>
<td>0.0225</td>
<td>0.0162</td>
<td>0.0185</td>
</tr>
<tr>
<td>$\omega_z$</td>
<td>0.0333</td>
<td>0.0253</td>
<td>0.0166</td>
</tr>
</tbody>
</table>

VI. Conclusions

This paper models spatial rigid motion of a space object using dual quaternions. Based on this model, an Extended Kalman Filter is implemented to estimate spatial motion in real-time. In particular, we track the projection of object feature points on an image plane and use the two dimensional tracking data to estimate the motion with six-degrees-of-freedom. Two distributed estimation approaches, dual decomposition and
Newton methods, are proposed to improve the estimation precision and convergence. The computational efficiency using a dual quaternion based model and improved estimation performance is demonstrated by simulation examples.

References


