Abstract—This paper proposes a distributed optimization algorithm for estimation of spatial rigid motion using multiple image sensors in a connected network. The objective is to increase the estimation precision of translational and rotational motion based on dual quaternion models and cooperation between connected sensors. The distributed Newton optimization method is applied to decompose the filtering task into a series of suboptimal problems and then solve them individually to achieve the global optimality. Our approach assumes that each sensor can communicate with its neighboring sensors to update the individual estimates. Simulation examples are demonstrated to compare the proposed algorithm with other methods in terms of estimation accuracy and converging rate.

Index Terms—Dual Quaternion; Distributed Estimation; Spatial Rigid Motion

I. INTRODUCTION

Spatial rigid motion with both translational and rotational evolutions has been found in a variety of dynamical systems, such as robotics, spacecraft, biomechanics, physics engines, and many others. Precise real-time motion estimation is critical for autonomous motion control when disturbances that require frequent tracking of the motion states are considered. The commonly used motion tracking instrument is the Inertial Navigation System (INS) equipped with a computer, motion sensors, and rotation sensors to continuously provide position, orientation, and velocity of a moving object [1]. However, when such an INS instrument is not available or the concerned object is not accessible for INS installation, real-time estimation of both translational and rotational motion of a dynamical system becomes a challenging task.

One of the challenges comes from the coupled translational and rotational motion, which requires a compact and efficient mathematical model to describe the combined motion. Nonlinearity of expressions describing translational and rotational motion makes the real-time estimation extremely difficult, especially when Euler angles are included in the rotational expression. Quaternions have been introduced as mathematical tools for calculating three-dimensional (3D) rotations to avoid singularity and reduce expensive computational load created by Euler angle expressions [2]. As alternative and powerful tools for representing object orientation, quaternions have been acknowledged as playing indispensable role in dynamical systems due to their unambiguous, unencumbered, and computationally-efficient features [3]. However, when both rotations and translations simultaneously occur in dynamical systems, quaternions alone cannot represent the spatial transformations. Thus, dual quaternions, an extension of quaternions, have been invented to unify the representation of rotational and translation motion within a single invariant coordinate frame. In many practical applications, such as robot arms and satellite attitude control, dual quaternions have demonstrated their advantages in terms of compactness, nonsingularity, and computational efficiency [4], [5]. In this paper, the dual quaternion kinematics is introduced to represent spatial rigid motion.

If traditional image-based sensors are used for motion tracking, a portion of the sensing data will not be available when the tracking points or lines move out of its viewing zone or when the sensor vision is blocked by interference. Such missing measurement information due to visual constraints or sensor malfunction creates difficulty in maintaining continuity and completeness of the tracking data. Furthermore, the complicated motion, the huge amount of observation data, and the resulting computational burden motivates the investigation of an efficient estimation algorithm for processing the observed data in real time. Existing work relating to rigid motion estimation using dual quaternion models applied an extended Kalman filter (EKF) to estimate translational and rotational motion using a single image sensor [6], [7]. Considering the challenging observational environment and the constraints of individual sensors, a distributed estimation scheme, in which the operation of each sensor is independent of others while cooperation among sensors is allowed, is more applicable to spatial rigid motion estimation of complex dynamical systems.

Distributed estimation technology has been commonly used in process control, signal processing and information systems [8]. A subset of these efforts have generally been focused on the integration of measurements from all the sensors into a common estimate without using a centralized processor. For example, literature in [9] contributes a great deal of work towards achieving an average consensus among distributed filters. By implementing a low-pass or band-pass filter, the estimates will reach an average consensus in a distributed manner. Continuous efforts relative to such types of work have been applied by extension to heterogeneous network systems and continuous Kalman filters, producing fast convergence of a decentralized consensus [10], [11]. In this paper, we propose to solve the distributed estimation problem by formulating it as an optimization problem with consensus constraints. Inspired by the rapid convergence of the Newton’s method in solving network utility maximization problems [12], [13], we propose to design a Newton-type distributed estimation algorithm aimed at increasing the rate of convergence.

The main contribution of the present paper is to design a multi-sensor framework for estimation of the spatial rigid motion based on dual quaternions. Furthermore, we propose a fast convergent distributed estimation algorithm to process the multi-sensor measurement data in a connected network.

The organization of the paper is as follows. We introduce the dual quaternion and its kinematics in §II. The single EKF algorithm for motion estimation based on dual quaternion kinematics is described in §III. The distributed Newton-type estimation algorithm is presented in §IV. Simulation examples demonstrating the feasibility and improved performance of the proposed approaches are detailed in §V. We conclude the paper with a few remarks in §VI.
II. DUAL QUATERNION AND KINEMATICS

A. Quaternion

The classical quaternion definition is

\[ \mathbf{q} = (q_0, \mathbf{q}) , \]  

where \( \mathbf{q} \in \mathbb{R}^3 \) and \( q_0 \in \mathbb{R} \) are the vector part and scalar part of the quaternion, respectively. In the following, we use notation \( \mathbb{H} \) to represent the set of quaternions. A unit quaternion \( \mathbf{q} \in \mathbb{H} \) with unit 2-norm can be used to represent a rotation of angle \( \theta \) about a unit axis \( \mathbf{n} \) in the form of

\[ \mathbf{q} = (\cos \frac{\theta}{2}, \mathbf{n} \sin \frac{\theta}{2}) . \]  

B. Dual Quaternion

Dual quaternions, introduced by Clifford [14], can present six-degrees-of-freedom rigid transformations by unifying translation and rotation into a single-state frame. Mathematically, dual quaternion is defined as

\[ \mathbf{d} = \mathbf{q} + \mathbf{q} \varepsilon , \]  

where \( \mathbf{q} \in \mathbb{H} \) denotes the real part, \( \mathbf{q} \mathbb{H} \) denotes the dual part, and \( \varepsilon \) is the dual unit with \( \varepsilon^2 = 0 \) but \( \varepsilon \neq 0 \). A spatial transformation including both translations and rotations can be expressed as

\[ \mathbf{d} = (\cos \frac{\theta}{2}, \mathbf{n} \sin \frac{\theta}{2}) \mathbf{t} \otimes \mathbf{q} , \]  

where \( \mathbf{t} = (0, \mathbf{i}) \in \mathbb{H} \) is a quaternion composed of the position vector \( \mathbf{i} \in \mathbb{R}^3 \) and a zero scalar part and \( \otimes \) represents the quaternion multiplication.

C. Dual Kinematics

We can find the kinematics of a spatial transformation in terms of dual quaternions, given as [6]

\[ \mathbf{d} = \frac{1}{2} \mathbf{\omega} \mathbf{\omega} . \]  

where \( \mathbf{\omega} = (0, \mathbf{\omega}) \in \mathbb{H} \) and \( \mathbf{\omega} = [\omega_x, \omega_y, \omega_z]^T \in \mathbb{R}^3 \) is angular velocity of the rotating body evaluated in the inertial frame. Based on the above derivation of \( \mathbf{d} \) and expression of \( \mathbf{q} \) in (2.4), we then find the derivative of dual part \( \mathbf{q} \), expressed as

\[ \mathbf{q} \mathbb{H} = \frac{1}{2} \mathbf{\omega} \mathbf{\omega} \mathbf{q} , \]  

where \( \mathbf{v} = (0, \mathbf{v}) \in \mathbb{H} \) is a quaternion composed of the velocity vector \( \mathbf{v} = [v_x, v_y, v_z]^T \in \mathbb{R}^3 \) in the inertial frame and a zero scalar part.

III. DEVELOPMENT OF SINGLE SENSOR EKF ALGORITHM BASED ON DUAL KINEMATICS

A. Introduction to Extended Kalman Filter

Consider a continuous nonlinear system with system dynamics \( \dot{x} = f(x) + w \) and a observation function \( z = h(x) + v \), where \( x \in \mathbb{R}^n \) are the states, \( z \in \mathbb{R}^m \) are the measurements, \( w \in \mathbb{R}^n \) is system noise with zero mean Gaussian sequence and covariance \( Q \in \mathbb{R}^{n \times n} \), and \( v \in \mathbb{R}^m \) is measurement noise with zero mean Gaussian sequence and covariance \( R \in \mathbb{R}^{m \times m} \). By discretizing and linearization, we can find the corresponding discrete linear system

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

where \( F_k = \frac{\partial f(x)}{\partial x} \bigg|_{x=x_k} \in \mathbb{R}^{n \times n} \) which is state transition matrix at time interval \( k \) can be calculated by taking the partial derivative about 1-step prediction \( x_k^- \) obtained prior to the input of observations. The discrete observation model is

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

where \( H_k = \frac{\partial h(x)}{\partial x} \bigg|_{x=x_k} \in \mathbb{R}^{m \times n} \) is the observation matrix. Different from standard Kalman filter which has constant state and observation matrices, the state and observation matrices in EKF are updated at each time interval \( k \).

The EKF algorithm for the system described by Eq. (3.7) and Eq. (3.8) finds the maximum-likelihood estimations \( \hat{x}_k \) by minimizing the following least-square objective function,

\[ J = \frac{1}{2} (x_k - x_k^-)^T(P_k^-)^{-1}(x_k - x_k^-) + \frac{1}{2}(z_k - H_k x_k)^T R^{-1}(z_k - H_k x_k) , \]  

where \( x_k^\hat{} \) is the prediction of the state before the measurements are made and \( P_k^- = E[(x_k - x_k^-)(x_k - x_k^-)^T] \) is the error covariance prior to incorporating the measurement at time \( k \) into the estimation. Given all the information up to time \( k \) in terms of the prediction state \( x_k^- \) and the measurement \( z_k \), the optimal solution of Eq. (3.9) provides a recursive estimation \( \hat{x}_k \) as following,

\[ \hat{x}_k = x_k^- + K_k(z_k - H_k x_k^-) , \]  

with Kalman gain expressed as \( K_k = P_k^- H_k^T R^{-1} \). In the Kalman gain expression is the covariance matrix of the error vector \( x_k - \hat{x}_k \), thus

\[ P_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \]  

\[ = ((P_k^-)^{-1} + H_k^T R^{-1} H_k)^{-1} (P_k^-)^{-1} + H_k^T R^{-1} H_k) , \]  

where \( P_k^- = F_k P_k^- F_k^T + Q \).

The key elements before applying EKF are to construct the state transition and observation models. State transition model derived from dual kinematics indicates the evolution from current to future states. Observation model using single image sensor can be constructed through the relationship between two relative frames, the object frame and the camera frame. Before introducing these models, we first select the following states to represent the spatial rigid motion of a concerned object. They are

\[ x = [\mathbf{q}^T \mathbf{q}^T \mathbf{\omega}^T \mathbf{v}^T]^T . \]  

B. State Transition Model Based on Dual Kinematics

By discretizing the dual kinematics expressed in Eq. (2.5) and Eq. (2.6) with sampling time \( \tau \), the state transition model associated with first eight elements of \( x \) becomes [15]

\[ \mathbf{q}_k = \mathbf{q}_k + \frac{\tau}{2} \mathbf{\omega}(k) \mathbf{q}_k(k) \]  

\[ \mathbf{q}_{k+1} = \mathbf{q}_{k+1} + \frac{\tau}{2} \mathbf{\omega}_k(k) \mathbf{q}_k(k) + \frac{\tau}{2} \mathbf{\omega}_k(k) \mathbf{q}_k(k) \]  

\[ + \frac{\tau}{2} (\mathbf{\omega}(k) \mathbf{q}_k(k) + \mathbf{\omega}_k(k) \mathbf{q}_k(k)) . \]  

By expanding the quaternion multiplication, we find the following relationship,

\[ \mathbf{\omega} \otimes \mathbf{q} = \mathbf{S}\mathbf{\omega} \]  

\[ \mathbf{v} \otimes \mathbf{q} = \mathbf{S}\mathbf{v} \]  

\[ \mathbf{t} \otimes \mathbf{q} = \mathbf{M}\mathbf{S}\mathbf{w} , \]  

where \( \mathbf{S} = \begin{bmatrix} \mathbf{q}_0 & -\mathbf{q}^T \mathbf{d} - K(q) \end{bmatrix} \) and \( \mathbf{M} = \begin{bmatrix} 0 & -\mathbf{i} \mathbf{K}(\mathbf{q}) \mathbf{^T} \end{bmatrix} \), the skew-symmetric matrix \( \mathbf{K}(\mathbf{q}) = \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix} \). In this paper, we
assume the linear and angular velocity are both constant. Based on
the relationship developed in Eq. (3.13), the state transition matrix
\( F(x_k) \) at discrete point \( k \) is expressed as
\[
F(x_k) = \begin{bmatrix}
    I & 0 & \frac{1}{2} S & 0 \\
    0 & I & \frac{1}{2} M S & \frac{1}{2} S \\
    0 & 0 & I & 0 \\
    0 & 0 & 0 & I
\end{bmatrix}_k.
\]
(3.14)

C. Observation Model

The measurement instruments used here to estimate the spatial
rigid motion of concerned object are image sensors/cameras. A set
of featured points of the object will be identified before observing
process. By recording the images of the featured points in the
camera frame, we aim at estimating both translational and rotational
motion of concerned object. Before we setup the observation
model, we first find out the relationship between the object frame
and the camera frame. Figure 1 demonstrates the coordinates of
one featured point in body frame \((x^c, y^c, z^c)\) and camera frame
\((x^r, y^r, z^r)\). The measurements of the object are obtained by
observing its projection on the image plane, denoted as \((x^i, y^i)\).

By dual multiplication, a dual quaternion transformation between
two frames is represented by
\[
\hat{p}^c = \hat{q} \hat{p}^r \hat{q}^*.
\]
(3.15)
where \( \hat{p}^c = 1 + (x^c i + y^c j + z^c k) \varepsilon \) and \( \hat{p}^r = 1 + (x^r i + y^r j + z^r k) \varepsilon \)
denote the dual representation of the featured point in camera
frame and object frame, respectively. \( \hat{q}^* \) represents the conjugate of dual
quaternion, \( \hat{q} \). Notation ‘\( \hat{\cdot} \)’ refers to dual quaternion multiplication.
The transformation process can be decomposed into two parts, pure
translation and pure rotation. Under this assumption, Eq. (3.15) can
be rewritten as
\[
\hat{p}^c = 1 + (q_r \otimes p^c_0) \otimes q^*_r + q_r \otimes q^*_r + q_d \otimes \hat{q}_r^* \varepsilon,
\]
(3.16)
where \( p^c_0 = (0, x^c i + y^c j + z^c k) \in \mathbb{H} \) represents the position
quaternion in object frame. From the above expression, we can find
the dual part of \( \hat{p}^c \), which indicates the position of the featured
point in camera frame, and rewrite it as
\[
p^c_0 = q_r \otimes p^c_0 \otimes q^*_r + 2 q_d \otimes q^*_r,
\]
(3.17)
where \( p^c_0 = (0, x^c i + y^c j + z^c k) \in \mathbb{H} \) represents the position
quaternion in camera frame. The first term in the right side of
Eq. (3.17) describes the rotation of the featured point and the
second term indicates its translation. Therefore, substituting \( q_d \) by
\((q_{10}, q_{11} + q_{23} k)\) and \( q_d \) by \((q_{00}, q_{11} + q_{23} k)\) in
Eq. (3.17) leads to
\[
\begin{bmatrix}
    0 \\
    x^o \\
    y^o \\
    z^o
\end{bmatrix} = R^r \begin{bmatrix}
    0 \\
    x^c \\
    y^c \\
    z^c
\end{bmatrix} + 2 \begin{bmatrix}
    q_{00} q_{10} + q_{11} q_{20} + q_{23} q_{30} \\
    q_{00} q_{11} - q_{11} q_{20} + q_{23} q_{31} - q_{33} q_{30} \\
    q_{00} q_{23} - q_{11} q_{23} + q_{23} q_{31} + q_{33} q_{30} \\
    q_{00} q_{30} - q_{11} q_{30} - q_{23} q_{31} + q_{33} q_{30}
\end{bmatrix},
\]
where \( R^r = \begin{bmatrix} 0_{1,3} & R^r \end{bmatrix} \) and \( R^r \) is a rotation matrix represented by
\( q_r \). Corresponding projection of the featured point on 2-D image
plane are obtained by [16]
\[
x^i = F_c x^r, y^i = F_c y^r,
\]
(3.18)
where \( F_c \) is the focal length of image sensor, \( P_x \) and \( P_y \) are inter-
pixel spacing along \( x \) and \( y \) axis on the image plane. Projection of
the featured points on the 2-D image plane is the measurement,
denoted as \( z \). In order to more precisely estimate the spatial
rigid motion of the concerned object, we track the projection of
three featured points on the image plane. In addition, we put unit
constraint on the real part of dual quaternion such that \( q_r^T q_r = 1 \), as well
as orthogonal constraint, \( q_r^T q_d = 0 \). Under these assumptions,
the components in the measurements are composed of
\[
z = \begin{bmatrix} x^i_1 & y^i_1 & x^i_2 & y^i_2 & x^i_3 & y^i_3 & q_r^T q_d & q_r^T q_c & q_r^T q_d \end{bmatrix}^T.
\]

According to findings in Eq. (3.18), the measurements are functions
of states \( x \), labeled as \( h(x) \). We then can find the discrete observation
matrix \( H_k \) in Eq. (3.8) according to \( H_k = \frac{\partial h(x)}{\partial x} \vert_{x=x^k} \).

By now, we have obtained the state transition and observation
models for estimation of spatial rigid motion using a single image
sensor. By applying the above described EKF algorithm, we can process
the observation data based on the developed models. However,
due to limited field-of-view of single sensor, the tracking
data of featured points may not be available during the observation
procedure. Therefore, a multi-sensor observation framework is
developed in this paper to improve the estimation precision.

IV. Newton-Type Distributed Estimation Algorithm

A. Multi-Sensor Networks and Problem Formulation

As we discussed in the introduction section, the disadvantage
of using single image sensor is that when the featured points
move out of the view zone, the measurement data will not be
available. The limitation of single image sensor will significantly
impact the estimation accuracy. Therefore, we propose a multi-
sensor network to track the object’s spatial motion from different
sensors simultaneously.

In a connected sensor network, we assume 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
connection which is defined by the entries of the adjacency matrix
\( A \) 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 finally. In such
system, the information filter or some other data fusion algorithm
that requires fully connected network cannot be applied in partially
connected system. In addition, the fully connected network requires
large data communication and storage which may bring difficulty
for implementation with the increase of data numbers. Furthermore,
without consensus constraints, there is no limitation to converge the
final result to the average consensus.
For a network system with $N$ sensors, adjacency matrix $A$ which is a symmetric matrix with zero diagonal entries indicates the neighbors for sensor $i$ by the off diagonal entries $A_{i,j}$, $i,j = 1, \ldots, N$, $i \neq j$. 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 the connected neighbors. By passing the identity request from one node to the other in the network, we can transfer the consensus request to the system. With the neighborhood consensus constraints on the estimates, we have the following relationships:

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

(4.19)

where $a_{i,j}$ denotes the element $A_{i,j}$ in matrix $A$. If there is connection between sensor $i$ and $j$, the consensus condition expressed in Eq. (4.19) will have $\hat{x}(i) = \hat{x}(j)$, otherwise such consensus constraint between node $i$ and $j$ does not exist. Since matrix $A$ is symmetric, we will have $a_{i,j} = a_{j,i}$.

As summary, the multi-sensor network estimation problem with consensus constraints at time step $k$ is formulated as

$$J = \min_{x_1, \ldots, x_N} \sum_{i=1}^{N} (x_i - x_i^*)^T (P_k^+)^{-1} (x_i - x_i^*) + (z_i - H_i(x_i)) H_i^T R^{-1} (z_i - H_i(x_i))$$

s.t. $\quad a_{i,j}\hat{x}(i) - a_{i,j}\hat{x}(j) = 0, \quad i = 1, \ldots, N, \quad i > j,$

(4.20)

where $N$ is the number of sensors in the connected network.

**B. 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 consensus based distributed estimation approach requires iterative coordination between networked sensors. For each iteration, the individual sensor must process the new estimates with the updated coordination variables using EKF. Even though the dual quaternion based models simplified the representation of spatial rigid motion, at least eight dual elements are included in the estimates for every iteration, so this process is time and resource consuming. 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 [12], [13], we propose a Newton-type distributed estimation algorithm, aiming at increasing the rate of convergence.

Consider the following constrained optimization problem

$$J = \min_{x_1, \ldots, x_n} \sum_{i=1}^{n} f_i(x_i)$$

s.t. $\quad Cx = b,$

(4.21)

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

$$x^{j+1} = x^j + s^j \Delta x^j,$$

(4.22)

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 equation:

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

(4.23)

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 (4.23), we get

$$\Delta x^j = -\left(\nabla^2 f^j + C^T w^j\right)^{-1} \left(\nabla f^j + C^T w^j\right),$$

(4.24)

$$C(\nabla^2 f^j + C^T w^j) = -C(\nabla^2 f^j + C^T w^j).$$

(4.25)

Since the objective function is decomposable in terms of $x(i)$, the diagonal elements in Hessian matrix can be calculated individually, denoted as $\nabla^2 f^j_{i}(i) = \partial^2 f_j/dx(i)^2$. However, finding $w^j$ requires global information to calculate the matrix $(C(\nabla^2 f^j + C^T w^j))^{-1}$. Work in [13] 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 consensus constraints formulated in (4.20) can be handled as one of the general constrained optimization problems described in (4.21). We use $\nabla f^j_{x(i)}$ and $\nabla^2 f^j_{x(i)}$ to represent the elements in the gradient vector and Hessian corresponding to $x(i)$ in iteration $j$. Their expressions in solving estimation problem of (4.20) are found as

$$\nabla f^j_{x(i)} = (P_k^+)^{-1} (x_i(i) - x_i^j(i)) \quad -H_k(i)^T R^{-1} (z(i) - H_k(i)x_i(i))$$

(4.26)

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

(4.27)

By substituting $\nabla f^j_{x(i)}$ and $\nabla^2 f^j_{x(i)}$ in (4.25), the Newton direction for updating $x(i)$ only is determined by

$$\Delta x^j(i) = -\left(\nabla^2 f^j_{x(i)}\right)^{-1} \left(\nabla f^j_{x(i)} - (\nabla^2 f^j)^{-1} C^T w^j\right).$$

The first term in the above equation is dependent on states and measurements of sensor $i$ only. Furthermore, $w^j$ can be solved by (4.25) through the decentralized method in [13]. 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 [13]. We firstly define two matrices,

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

(4.28)

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

(4.29)

Let $w(0)$ be an arbitrary initial vector and the sequence $w(s)$ is generated by

$$w(s+1) = (D^j)^{-1} (-C(\nabla^2 f^j)^{-1} \nabla f^j)$$

(4.30)

$$- (D^j)^{-1} B^j w(s).$$

When $s \to \infty$, $w(s)$ in (4.25) is obtained.

**V. Simulation Example**

Figure 2 demonstrates the layout of image sensors and trajectories of the tracking points on the moving object. To make it simple, we use two image sensors which can communicate with each other. Sensor 1 is set 10 meters directly below the object initial position. Sensor 2 is set at the left hand side of the object initial position with 10 meters shifting along $-y^j$ axis and a rotation of $90^\circ$ with respect to $-x^j$ axis. Three points on surface of the object are selected as featured points. They are located on the axis of object frame and are one meter away from the origin. The object is moving along positive $x^j$ axis with constant linear velocity $v_x = 1m/s$. Simultaneously, it is rotating around axis $x_0$ with a constant angular velocity $\omega_z = 2\pi rad/s$. Table I shows relative parameters in the observing model. The measurement noise considered when observing the projections in the image plane is Gaussian white noise with 0.06 pixel$^2$ variance.
There are two important parameters involved in EKF: process noise covariance matrix Q and measurement noise covariance matrix R. In our model, the object in the simulation has constant linear and angular velocity. It is expected that there is no dramatic change in the system states. Therefore, it is reasonable to use a constant matrix Q, which will not significantly affect the tracking accuracy. In the simulation, Q is set as a diagonal matrix. The diagonal elements corresponding to dual quaternion states in Q are set as 0.01. The elements related to angular and linear velocity are set as 0.1. Another parameter is the measurement noise covariance matrix R. Since we consider white noise in the observation model, R is assumed to be a diagonal matrix with 0.06 in diagonal elements, as shown in Table I.

Before we proceed with EKF, two parameters, the initial states and the covariance matrix of the estimation error, need to be initialized at the beginning. To demonstrate the effectiveness of the proposed distributed estimation algorithm, offsets are added to real initial states. For example, the elements in initial position vector have offset of ±0.1 meters, the elements in linear and angular velocity vectors have offset of −1 and −2π, respectively. Correspondingly, we set up the initial covariance matrix of the estimation error, \( P_0 \), as a diagonal matrix according to the definition, \( P_0 = E[(x_0 - x_0^*)(x_0 - x_0)^T] \). It is expected that even with the large offsets of initial states, the proposed distributed estimation algorithm can eventually converge to the real states.

Table I.

<table>
<thead>
<tr>
<th>Parameter Settings</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel spacing along x, ( P_x )</td>
<td>2 \times 10^{-3} \text{m/pixel}</td>
</tr>
<tr>
<td>Pixel spacing along y, ( P_y )</td>
<td>2 \times 10^{-3} \text{m/pixel}</td>
</tr>
<tr>
<td>Sample period, ( T )</td>
<td>0.02 s</td>
</tr>
<tr>
<td>Focal length, ( P_c )</td>
<td>0.017 m</td>
</tr>
<tr>
<td>Measurement noise variance</td>
<td>0.06 \text{pixel}^2</td>
</tr>
</tbody>
</table>

Based on the above simulation scenario, two simulation examples are given below. The first implements EKF to estimate the time history of dual quaternions, angular and linear velocities using individual image sensors, where no cooperation is considered between the sensors. The second applies the proposed Newton-type distributed estimation algorithm, where cooperation between the connected sensors are considered. Figures 3-6 demonstrate real states (red) and estimated states (blue and black) from two individual sensors. The estimates obtained from sensor 1 are in black color and other sets from sensor 2 are in blue color. The average mean square error (MSE) from two sensors is listed in Table II.

Table II.

<table>
<thead>
<tr>
<th>Angular Velocity</th>
<th>MSE</th>
<th>Linear Velocity</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \omega_x )</td>
<td>0.0071</td>
<td>( \nu_x )</td>
<td>0.2801</td>
</tr>
<tr>
<td>( \omega_y )</td>
<td>0.0060</td>
<td>( \nu_y )</td>
<td>0.3670</td>
</tr>
<tr>
<td>( \omega_z )</td>
<td>0.0045</td>
<td>( \nu_z )</td>
<td>0.3122</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

This paper models spatial rigid motion using dual quaternions. Based on this model, extended Kalman Filter is implemented to
estimate spatial motion in real-time. In particular, we track the projection of object featured points on image plane and use the two dimensional tracking data to estimate the motion with six-degree-of-freedom. A new distributed estimation approach based on distributed Newton method is proposed to improve the estimation precision and convergence. The calculation efficiency using dual quaternion based model and improved estimation performance is verified by simulation examples.

REFERENCES