Dual Quaternion Visual Servo Control

Ryan Saltus, Iman Salehi, Ghananeel Rotithor, and Ashwin P. Dani

Abstract— **This paper focuses on a dual quaternion-based estimation and control approach for position-based visual servoing (PBVS). The pose estimation of the camera is achieved using a dual quaternion-based Extended Kalman Filter (EKF), which estimates the position and orientation of the camera based on feature points acquired through a sequence of camera images. Based on the estimation, a dual quaternion control law is developed to regulate the camera to the desired pose. Leveraging the local exponential stability of the EKF and the global exponential stability of the designed controller, a nonlinear separation principle is used to prove the stability of the joint estimation and control for PBVS. The method is distinguished from other PBVS methods in the sense that a compact representation of dual quaternion is used to represent the pose, and a joint stability of estimator and controller for PBVS in dual quaternion space is presented. The proposed dual quaternion PBVS method is validated using a simulation.**

I. Introduction

Visual servo control refers to the use of camera image data to control the motion of a camera attached to a robot [1]. Different approaches exist in the literature, including image-based, position-based, and combined approaches [2], [3], [4], [5], [6]. Position-based visual servo (PBVS) control is a closed-loop control scheme that uses camera image feedback to estimate the relative pose (position and orientation) between the camera and an object reference frame, which is used to control the motion of a robot observing the static object or scene. In order to compute pose, the 3-D model of the observed object must be known if a single image is available [7]. If the 3D model of the object is not known, multiple camera views can be used to estimate the feature point depth which in turn can be used to estimate the pose of the object, see, e.g., [8]. If a single length between two feature points is known then the camera pose can also be estimated using the method presented in [9].

Visual servo control has been an active topic of research within robotics. In [10], a methodology for Cartesian-based visual servoing is proposed, and an EKF is used to estimate the relative pose of the object in the camera frame. PBVS under dynamic visibility checking of feature points is proposed in [11] to check if the features stay in the field of view (FOV). Pose estimation is an essential step for the PBVS controller. Several advances have been made in pose estimation algorithms. In [12] a nonlinear method that consists of minimizing a cost function using iterative algorithms is developed to estimate the pose from image feature points. In [13], the QuEst algorithm is presented, which estimates the position and orientation separately from a minimal number of feature points. These pose estimation algorithms are geometric in nature that use geometric properties to solve the problem. Other pose estimation algorithms exist where the pose is represented using different parametrizations such as Rodrigues' angle-axis parametrization or homogenous transformation matrices. Quaternions have been used in an adaptive homography-based visual tracking controller as proposed in [14], which uses a quaternion formulation to represent the tracking error. Dual quaternion representation provides a compact, accurate and computationally efficient representation over these parametrizations [15]. However, to the authors' knowledge a dual quaternion representation has not been used for visual servo control so far.

Dual quaternions are an extension of the dual numbers to include quaternion algebra. Dual quaternions are especially useful for representing rigid Euclidean transformations by providing a convenient, compact, and meaningful way of describing position and orientation. Requiring at most 8 numbers for a full pose description, dual quaternions retain many desirable properties of quaternions, including computational efficiency and freedom from singularities. Representing pose in this way has benefits over the traditional transformation matrix because of the reduced memory requirements to represent pose (8 numbers vs 12), as well as reduction in computational cost [15]. A dual quaternion-based asymptotically stable observer for estimating velocities is designed in the context of a leader follower system along with proofs of stability in [16]. The designed observer is used in conjunction with a proportional-derivative (PD) controller using a strictification of the Lyapunov function to ensure a separation property between the observer and controller. In [17], dual quaternions have been used to model kinematics of soft continuum robots/actuators, which typically require infinite dimensional representation, see e.g., in [18], [19]. A method for contour-based rigid body tracking with simultaneous camera calibration using a dual quaternion-based extended Kalman filter (EKF) is developed in [20].

This paper presents a dual quaternion-based visual servo controller in which the relative pose between the camera and the reference frame is estimated using a dual quaternion

This work was supported in part by Office of Naval Research Award No. N00014-20-1-2040, in part by Space Technology Research Institutes grant (number 80NSSC19K1076) from NASA Space Technology Research Grants Program, and in part by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award Number DE-EE0007613.

Ryan Saltus, Iman Salehi, Ghananeel Rotithor, and Ashwin P. Dani are with the Department of Electrical and Computer Engineering at University of Connecticut, Storrs, CT 06269. Email: {ryan.saltus; iman.salehi; ghananeel.rotithor; ashwin.dani}@uconn.edu.

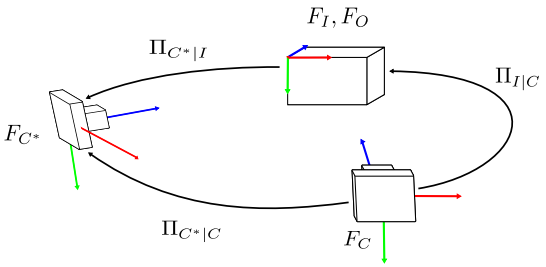

Fig. 1. Reference frames and transformations involved in the PBVS task.

representation of pose, followed by a PBVS control design that utilizes the estimated pose to regulate the current camera pose to a desired pose. Classical PBVS methods typically suffers from instability issues due to feature noise. In this paper, a locally exponentially converging continuous-time extended Kalman filter (EKF) is presented to estimate the relative pose. In the EKF, a dual quaternion representation is used as the state and image feature points are used as measurements. Then, a globally exponentially stable pose regulation controller on dual quaternion space is proposed, followed by a joint estimator-controller stability analysis of the PBVS controller on dual quaternion space. The joint estimator-controller stability analysis provides robustness guarantees against the image noise and other disurbances introduced due to estimation errors in the transients, which mitigates the instability issues observed in the PBVS method.

II. Preliminaries

A. Perspective Projection Model and PBVS

Consider a moving camera observing the feature points of a static object. Let F_C be the coordinate frame attached to the camera such that the Z axis of the frame is collinear with the optical axis of the camera, and the origin of F_C is at the principal point. Let F_O be the static coordinate frame attached to the origin of the object. For ease of notation, the inertial coordinate frame F_I is assigned to coincide with the object coordinate frame F_O. The objective of PBVS is to move the camera to a desired position and orientation F_{C^*} while utilizing the image feedback of the camera. A summary of coordinate frame relationships can be seen in Fig 1.

It is assumed that the positions of the feature points with respect to F_I are known. The camera can be modeled using the common perspective projection model as

$$\begin{bmatrix} \lambda u_i \\ \lambda v_i \\ \lambda \end{bmatrix} = \begin{bmatrix} fs_u & 0 & o_u \\ 0 & fs_v & o_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{I|C} & \mathcal{T}_{I|C} \end{bmatrix} \begin{bmatrix} X_{i|I} \\ Y_{i|I} \\ Z_{i|I} \\ 1 \end{bmatrix}$$

$$z = A\Pi_{I|C} \begin{bmatrix} N_{i|I} \\ 1 \end{bmatrix}, \tag{1}$$

where A is the camera calibration matrix, λ is a scaling factor, $\Pi_{I|C}$ is a matrix consisting of the rotation matrix $R_{I|C} = \left[r_x^T, r_y^T, r_z^T\right]^T \in \mathbb{R}^{3\times 3}$ with $r_x, r_y, r_z \in \mathbb{R}^{1\times 3}$ and translation vector $\mathcal{T}_{I|C} = [\mathcal{T}_x, \mathcal{T}_y, \mathcal{T}_z]^T \in \mathbb{R}^3$ of the inertial frame with respect to the camera frame, and $N_{i|I} \in \mathbb{R}^3$ is the coordinates of the ith feature point in the inertial frame. u_i and v_i are the pixel locations of the feature point, f is the focal length of the camera, s_u and s_v are the dimensions of the pixel in the u and v direction, and o_u and o_v are the location of the principal point in pixels. For the remainder of the paper, coordinate frame transformations will be of the form $H_{A|B}$, where the transformation H is of the A frame with respect to the B frame. Velocities will be of the form $V^C_{A|B}$, where the velocity V is of the A frame with respect to the B frame, expressed in the C frame.

B. Quaternions and Dual Quaternions

A *quaternion* is an extension of the complex numbers to $\mathbb{R}^4$, consisting of a scalar part and three imaginary parts. A quaternion q can formally be defined as $q = q_w + q_i i + q_j j + q_k k$, where $i^2 = j^2 = k^2 = -1$, $ij = k$, $jk = i$, and $ki = j$. Convenient shorthands include $q = (q_w, q_v)$ or $\check{q} = \begin{bmatrix} q_w & q_v{}^T \end{bmatrix}^T \in \mathbb{R}^4$, where $q_w \in \mathbb{R}$ is the scalar part and $q_v = \begin{bmatrix} q_i & q_j & q_k \end{bmatrix}^T \in \mathbb{R}^3$ is the vector part. Some common quaternion operations are defined for quaternions q and p as

$$q + p = (q_w + p_w, q_v + p_v) \tag{2}$$

$$kq = (kq_w, kq_v), k \in \mathbb{R} \tag{3}$$

$$qp = (q_w p_w - q_v \cdot p_v,\ q_w p_v + p_w q_v + q_v \times p_v) \tag{4}$$

$$\bar{q} = (q_w, -q_v) \qquad \|q\| = \sqrt{\check{q} \cdot \check{q}} \tag{5}$$

where $\bar{q}$ is the quaternion conjugate. A *dual number* is defined by $\mathbf{d} = a + \epsilon b$, where a is the real part and b is the dual part. ϵ is known as the *dual operator*, satisfying the properties $\epsilon^2 = 0, \epsilon \neq 0$. A *dual quaternion* is a dual number with quaternions as elements, and is defined as $\mathbf{q} = q_r + \epsilon q_d$, where q_r is the real part of dual quaternion and q_d is the dual part. A *dual vector quaternion* is a dual quaternion with 0 scalar elements. Some common dual quaternion operations are defined as

$$\mathbf{q} + \mathbf{p} = (q_r + p_r) + \epsilon (q_d + p_d) \tag{6}$$

$$k\mathbf{q} = (kq_r) + \epsilon (kq_d) \tag{7}$$

$$\mathbf{qp} = (q_r p_r) + \epsilon (q_d p_r + q_r p_d) \tag{8}$$

$$\bar{\mathbf{q}} = (\bar{q}_r) + \epsilon (\bar{q}_d) \tag{9}$$

where $\bar{\mathbf{q}}$ is the dual quaternion conjugate.

A dual quaternion can be used to define the rigid body transformation between reference frames. Unit quaternions ($\|q\| = 1$) provide a non-minimal singularity free representation of rotation, and can be used to describe the frame rotation around a unit axis n with angle $|\theta|$ as

$$q = \left(\cos\left(\frac{|\theta|}{2}\right), \sin\left(\frac{|\theta|}{2}\right) n\right) \tag{10}$$

Extending this concept, a unit dual quaternion can be used to represent a combined rotation and translation simultaneously. A frame transformation $\mathbf{q}$, composed of a translation $\mathcal{T}$ written as a vector quaternion $T = (0, \mathcal{T})$ followed by a rotation q, can be represented as

$$\mathbf{q} = q + \frac{\epsilon}{2}(Tq) \tag{11}$$

To be a valid representation of a translation and rotation, the unit dual quaternion must satisfy the unity constraints

$$\|q_r\| = 1, \quad \bar{q}_r q_d + \bar{q}_d q_r = 0 \tag{12}$$

The logarithm of a pose dual quaternion is defined as [21]

$$\log(\mathbf{q}) = \frac{1}{2}(\theta_q + \epsilon T) \tag{13}$$

where $\theta_q = (0, |\theta| n)$.

The adjoint transformation is a transformation that switches the reference frame of a dual vector quaternion in which the coordinates are expressed. Given a dual vector quaternion $\omega^B_{A|B}$ written in B coordinates, the adjoint transformation to represent that same dual vector quaternion in A coordinates is written as $\omega^A_{A|B} = \mathbf{q}_{A|B}\omega^B_{A|B}\bar{\mathbf{q}}_{A|B}$, where $\mathbf{q}_{A|B}$ is the transformation of the A frame with respect to the B frame.

III. Dual Quaternion Model Development

To estimate the relative pose between F_C and F_I, a dual quaternion-based EKF is used. A controller is developed that uses the estimated pose to regulate the current pose to the desired pose. This section describes the state dynamics and measurement models for the dual quaternion parametrization of the camera pose.

A. System Dynamics Model

Let $\mathbf{q}_{I|C}(t)$ be the dual quaternion parametrization of the pose transformation of the inertial frame with respect to the camera frame at time instant t. For ease of notation, the dependency of $\mathbf{q}_{I|C}(t)$ is abbreviated by $\mathbf{q}_{I|C}$, unless necessary for clarity. The dynamics of the dual quaternion $\mathbf{q}_{I|C}$ can be computed by taking the time derivative of (11) as

$$\dot{\mathbf{q}}_{I|C} = \frac{1}{2}q_{I|C}\omega^I_{I|C} + \epsilon\left(\frac{1}{2}q_{I|C}v^I_{I|C} + \frac{1}{4}T_{I|C}q_{I|C}\omega^I_{I|C}\right),$$

which can be broken down into

$$\dot{q}_{I|C,r} = \frac{1}{2}q_{I|C}\omega^I_{I|C} \tag{14}$$

$$\dot{q}_{I|C,d} = \frac{1}{2}q_{I|C}v^I_{I|C} + \frac{1}{4}T_{I|C}q_{I|C}\omega^I_{I|C} \tag{15}$$

with $\omega^I_{I|C}$ and $v^I_{I|C}$ defined as the angular and linear velocity vector quaternions. In order to facilitate the subsequent model development, a state space representation of the pose dynamics in (14)-(15) is used. Using the shorthand notation defined in Section II-B, the state vector is given by

$$x(t) = \left[\left(\check{q}_{I|C,r}\right)^T, \left(\check{q}_{I|C,d}\right)^T\right]^T, \tag{16}$$

where $\check{q}_{I|C,r} \in \mathbb{R}^4$ is the real part of the pose dual quaternion and $\check{q}_{I|C,d} \in \mathbb{R}^4$ is the dual part. The state dynamics equation can be written as

$$\dot{x}(t) = \phi(x(t))\,\xi(t) \tag{17}$$

where $x(t) \in \mathbb{R}^8$ is the state, $\phi(\cdot) : \mathbb{R}^8 \to \mathbb{R}^{8\times 8}$, and $\xi = \left[\left(\check{\omega}^C_{C|I}(t)\right)^T, \left(\check{v}^C_{C|I}(t)\right)^T\right]^T \in \mathbb{R}^8$ is the control input. Using the matrix form of quaternion multiplication, $\phi(\cdot)$ is defined as

$$\phi(x(t)) = \begin{bmatrix} \frac{1}{2}Q_r(t) & 0_{4\times 4} \\ \frac{1}{4}Q_d(t) & \frac{1}{2}Q_r(t) \end{bmatrix}\Psi(t), \tag{18}$$

where

$$\Psi(t) = -\begin{bmatrix} Q_r(t) & 0_{4\times 4} \\ Q_d(t) & Q_r(t) \end{bmatrix}\begin{bmatrix} \tilde{Q}_r(t) & 0_{4\times 4} \\ \tilde{Q}_d(t) & \tilde{Q}_r(t) \end{bmatrix} \tag{19}$$

is a matrix multiplication representation of the adjoint transform from reference frame C to I on $\xi(t)$. The dual quaternion multiplication matrix $Q(t)$ for the quaternion $q = q_w + q_i i + q_j j + q_k k$ and is given by

$$Q(t) = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix} \tag{20}$$

$$\tilde{Q}(t) = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & -q_w & q_z & -q_y \\ q_y & -q_z & -q_w & q_x \\ q_z & q_y & -q_x & -q_w \end{bmatrix} \tag{21}$$

and Q_r and Q_d in (18), (19) are of the real part and dual part of $\mathbf{q}_{I|C}$, respectively.

B. Measurement Model

In order to develop the continuous time measurement model as a function of the state, the equations in (1) are rewritten as

$$\begin{bmatrix} \lambda u_i \\ \lambda v_i \\ \lambda \end{bmatrix} = \begin{bmatrix} fs_u r_x N_{i|I} + fs_u \mathcal{T}_x + o_u r_z N_{i|I} + o_u \mathcal{T}_z \\ fs_v r_y N_{i|I} + fs_v \mathcal{T}_y + o_v r_z N_{i|I} + o_v \mathcal{T}_z \\ r_z N_{i|I} + \mathcal{T}_z \end{bmatrix}, \tag{22}$$

where $r_x, r_y, r_z, \mathcal{T}_x, \mathcal{T}_y, \mathcal{T}_z$ are functions of $\mathbf{q}_{I|C}$. The argument $\mathbf{q}_{I|C}$ has been omitted from (22) for brevity. The rotation portions can be directly calculated from the real part of the dual quaternion $q_{I|C,r}$, and the translational portion can be computed from the equation $T_{I|C} = 2q_{I|C,d}\bar{q}_{I|C,r}$. Let $y_i(t)$ be the pixel coordinates of the i^{th} image feature point, the measurement equation is written as

$$y_i(t) = h_i(x(t)) = \begin{bmatrix} u_i(t) \\ v_i(t) \end{bmatrix} = \begin{bmatrix} \frac{\lambda u_i(t)}{\lambda} \\ \frac{\lambda v_i(t)}{\lambda} \end{bmatrix} \tag{23}$$

The measurement model for n feature points is constructed by augmenting each y_i as

$$y(t) = h(x(t)), \tag{24}$$

where $y = \left[y_1^T, \cdots, y_n^T\right]^T \in \mathcal{F}$ such that $\mathcal{F} \subset \mathbb{R}^{2n}$ is a bounded set. To linearize the nonlinear measurement model in (23), we take the Jacobian with respect to the state vector. This can be done by direct differentiation of (23)

$$\begin{bmatrix} \frac{\partial u_i}{\partial x} \\ \\ \frac{\partial v_i}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\frac{\partial(\lambda u_i)}{\partial x}}{\lambda} - \frac{\frac{\partial \lambda}{\partial x}\lambda u_i}{\lambda^2} \\ \\ \frac{\frac{\partial(\lambda v_i)}{\partial x}}{\lambda} - \frac{\frac{\partial \lambda}{\partial x}\lambda v_i}{\lambda^2} \end{bmatrix} \tag{25}$$

The standard EKF equations are used to estimate the inertial frame with respect to the original frame. It is important to note that after each state update, a post normalization step must take place in order to ensure that the unit dual quaternion constraints in (12) are still satisfied.

IV. Continuous-Time EKF for Pose Estimation

Given the state dynamics in (17) and the measurement model in (24), the observer is given by

$$\dot{\hat{x}}\left(t\right)=\phi\left(\hat{x}\left(t\right)\right)\hat{\xi}\left(t\right)+W\left(t\right)\left(y\left(t\right)-h\left(\hat{x}\left(t\right)\right)\right), \tag{26}$$

where $W(t)$ is the filter-gain given by

$$W\left(t\right)=P\left(t\right)H\left(t\right)^{T}R^{-1}, \tag{27}$$

where $H\left(t\right)=\frac{\partial h(x)}{\partial x}$ is evaluated at $\hat{x}(t)$ and $P(t)$ is the solution of the differential Riccati equation

$$\dot{P}(t)=AP+PA^{T}+Q-PH^{T}R^{-1}HP,\quad P(t_{0})=P_{0} \tag{28}$$

where $A\left(t\right)=\frac{\partial(\phi(x)\xi)}{\partial x}$ is evaluated at $\hat{x}(t)$ and $\hat{\xi}(t)$. The constant matrices P_0, R, and Q are symmetric and positive definite. It is crucial for validity of the EKF to verify, the solution $P(t)$ of (28) exists for all $t > t_0$ and be upper and lower bounded by $\alpha_1 I \leq P(t) \leq \alpha_2 I$ for some positive constants α_1 and α_2. This can be verified if the pair $(A(t), H(t))$ is uniformly observable and both $A(t)$ and $H(t)$ are bounded. Lemma 1 proves that the continuous EKF is an exponentially stable observer for the estimation error defined by $\tilde{x}(t) = x(t) - \hat{x}(t)$.

Lemma 1: Given that the solution $P(t)$ of (28) exists and is bounded, then equilibrium $\tilde{x} = 0$ is exponentially stable and there exists positive constants c, k and β such that

$$\|\tilde{x}\left(0\right)\|\leq c\implies\|\tilde{x}\left(t\right)\|\leq ke^{-\beta(t-t_{0})},\quad\forall t>t_{0}>0$$

Proof: See [22] ∎

Remark 1: The domain of convergence of the EKF is given by $\Omega_{\text{EKF}}=\left\{\hat{x}\left(t\right)\in\mathbb{R}^{8}\middle|\|\tilde{x}\left(0\right)\|\leq c\right\},\quad\forall t>t_{0}>0$.

V. Control Development and Stability Analysis

Define the dual quaternion error $\tilde{\mathbf{q}} = \bar{\mathbf{q}}_{C^*|I}\bar{\mathbf{q}}_{I|C}(t)$. The dynamics of the dual quaternion error are given as

$$\dot{\tilde{\mathbf{q}}}=\frac{1}{2}\tilde{\mathbf{q}}\xi_{C|I}^{C} \tag{29}$$

The control objective is to regulate the current estimated pose $\hat{\mathbf{q}}_{C|I}$ to the desired pose $\mathbf{q}_{C^*|I}$ such that the logarithmic error

$$2\left\|\log\left(\bar{\mathbf{q}}_{C^{*}|I}\bar{\hat{\mathbf{q}}}_{I|C}(t)\right)\right\|\to0,\quad\text{as }t\to\infty \tag{30}$$

The control objective (30) can be achieved using a dual quaternion logarithmic controller of the form

$$\hat{\xi}_{C|I}^{C}(t)=-2k_{p}\log\left(\bar{\mathbf{q}}_{C^{*}|I}\bar{\hat{\mathbf{q}}}_{I|C}(t)\right), \tag{31}$$

where $\hat{\xi}_{C|I}^{C}(t)$ is the control velocity vector of the camera, log is the dual quaternion logarithm, and $k_p \in \mathbb{R}^+$ is a positive scalar gain.

Lemma 2: The nominal logarithmic controller

$$\xi_{C|I}^{C}(t)=-2k_{p}\log\left(\bar{\mathbf{q}}_{C^{*}|I}\bar{\mathbf{q}}_{I|C}(t)\right) \tag{32}$$

globally exponentially stabilizes the error dynamics in (29).

Proof: Based on the control objective, define logarithmic error $\mathbf{r} = \log(\tilde{\mathbf{q}})$ such that $\mathbf{r} \to 0$ as $\tilde{\mathbf{q}} \to \check{\mathbf{1}}$, where $\check{\mathbf{1}} = [1,0,0,0,0,0,0,0]^T$. Consider a candidate Lyapunov function defined by $V = \frac{1}{2} < \mathbf{r}, \mathbf{r} >$, where $< \cdot, \cdot >$ is the inner product on the dual vector space. Using Lemma 1 in [21], the time derivative of the Lyapunov function can be written as

$$\dot{V}=2<\mathbf{r},\xi_{C|I}^{C}> \tag{33}$$

Substituting the designed controller in (32), the derivative is simplified to $\dot{V} =< 2\mathbf{r}, -2k_p\mathbf{r} >= -\kappa V$, where $\kappa \in \mathbb{R}^+$. Thus, the control law ensures exponential stability of the dynamics in (29), which implies $2\left\|\log\left(\bar{\mathbf{q}}_{C^*|I}\bar{\mathbf{q}}_{I|C}(t)\right)\right\| \to 0$ and $x(t) \to x^*$. ∎

The control law in (32) and the EKF developed in Sections III and IV are used in a joint estimation and control scheme to regulate the camera pose to the desired pose. The controller in (32) is globally exponentially stable and the EKF is locally exponentially converging; thus, the system Θ defined in (34) can be proven to be asymptotically stable using a nonlinear separation principle.

Theorem 1: Consider the following system

$$\Theta:\begin{cases}\dot{x}\left(t\right)=\phi\left(x\left(t\right)\right)\hat{\xi}(t)\\ \dot{\hat{x}}\left(t\right)=\phi\left(\hat{x}\left(t\right)\right)\hat{\xi}(t)+W\left(t\right)\left[y\left(t\right)-h\left(\hat{x}\left(t\right)\right)\right]\\ \dot{P}(t)=A(t)P(t)+P(t)A^{T}(t)+Q\\ \qquad -P(t)H^{T}(t)R^{-1}H(t)P(t)\end{cases} \tag{34}$$

Under the assumption that the controller $\xi(t)$ is globally exponentially stable, which implies the state $x(t)$ in Θ remains in a compact set Ω that has the equilibrium point of the nominal controller $\forall t > 0$, the system Θ is asymptotically stable $\forall x(0) \in \Omega$, $\forall \hat{x}(0) \in \Omega_{\text{EKF}}$, and $\forall P(0) > 0$ on $\Omega \times \Omega_{\text{EKF}} \times P^+$.

Proof: See proof of Theorem 3 in [23] ∎

The proof of Theorem 1 is based on the fact that for a semi-trajectory $\Lambda = \{(\tilde{x}(t) = x - \hat{x}, \hat{x}(t), P(t)), t > 0\}$ of the controller-observer pair, there exists a nonempty ω-limit set of Λ, and the element $[0, x^*, P^*]$ belongs to the limit set. From the local exponential stability of the EKF, it can be concluded that Λ enters the basin of attraction of $[0, x^*, P^*]$ in a finite time. Based on the fact that the estimation error is locally exponentially stable and the controller is globally exponentially stable, it can be concluded that Θ is globally asymptotically stable.

VI. Simulation Results

A simulation is carried out to verify the performance of the dual quaternion estimator-controller for PBVS control in MATLAB 2020a using resources from the Visual Servoing

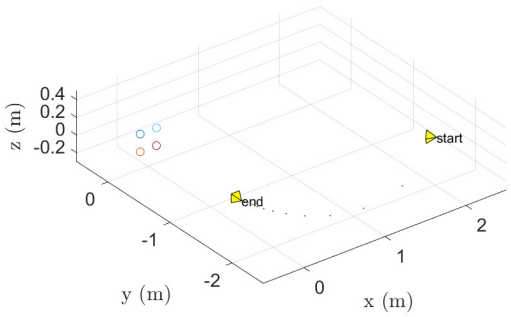

Fig. 2. Trajectory of the camera over time.

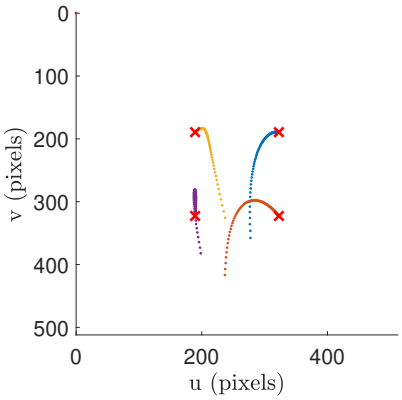

Fig. 3. The evolution of the feature points in the camera image plane. The feature points as viewed from the desired camera pose are displayed as red X's.

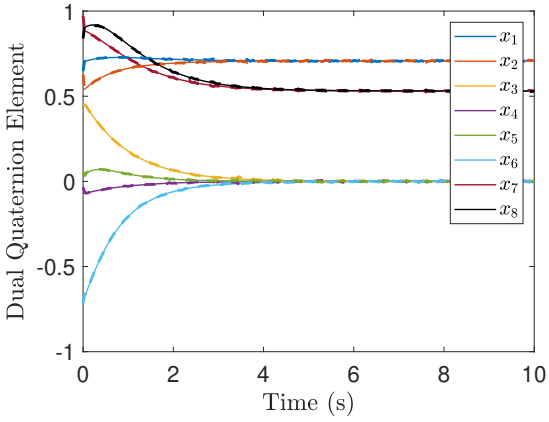

Fig. 4. The evolution of the state $(\check{\mathbf{q}}_{I|C})$. The actual state is shown as a solid line, and the corresponding filter estimate is shown in the same colored dotted line.

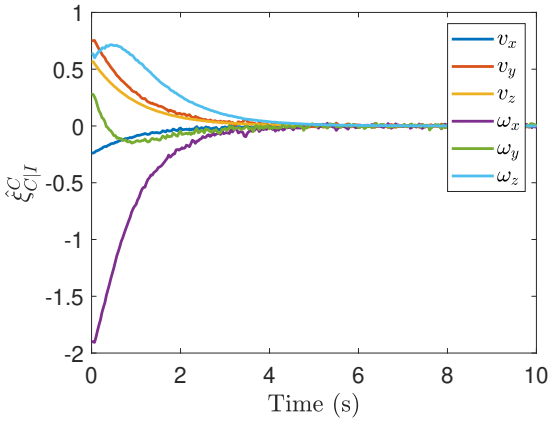

Fig. 5. The applied velocity over time. Linear velocities are in $\frac{\text{m}}{\text{s}}$ and angular velocities are in $\frac{\text{rad}}{\text{s}}$. All velocities are represented in the camera frame.

Toolbox from [24]. The simulation is performed over a period of 10 seconds, at a frequency of 30 frames per second. The image resolution is set to be 512×512, the center pixel (o_u, o_v) is set to $(256, 256)$, and the focal length $fs_u = fs_v = 1000\text{px}$. Additive white Gaussian noise with a signal to noise ratio (SNR) of 20dB is added to all of the measurements taken by the camera. The object frame is chosen to coincide with the inertial frame, and the object is modeled as four coplanar points with coordinates $N_{1|I} = [0.1, 0, 0.1]^T$, $N_{2|I} = [-0.1, 0, 0.1]^T$, $N_{3|I} = [-0.1, 0, -0.1]^T$, and $N_{4|I} = [0.1, 0, -0.1]^T$. The camera is initialized at the pose

$$\check{\mathbf{q}}_{I|C}(t_0) = [0.7036, 0.5408, 0.4555, -0.0706, 0.0412, -0.6813, 0.8853, 0.9043]^T,$$

and the desired camera pose is set to

$$\check{\mathbf{q}}_{I|C^*} = [0.7071, 0.7071, 0, 0, 0, 0, 0.5303, 0.5303]^T$$

The dual quaternion-based EKF is initialized close to the starting point with a random positional and rotational disturbance applied to simulate initial estimation error. The constant matrices are initialized as $P_0 = 0.1^2 \cdot \mathbb{I}_8$, $Q = 0.1^2 \cdot \mathbb{I}_8$, and $R = 1^2 \cdot \mathbb{I}_8$. The controller gain is set to $k_p = 1$, and is tuned experimentally.

The results of the simulation are summarized in Figs. 2-7. In Fig. 2, the time evolution of the camera pose and the 3D locations of the tracked feature points are displayed. The evolution of the pixel values corresponding to each feature point over time can be seen in Fig. 3. As seen in Fig. 3, the points clearly converge to the desired pixel locations marked in red X's. Fig. 4 shows the pose evolution over time, and Fig. 5 shows the applied velocity (in units of $\frac{\text{m}}{\text{s}}$ and $\frac{\text{rad}}{\text{s}}$) with respect to time. In Fig. 6, the logarithmic error between the estimated pose and the actual pose is seen converging, which validates the local exponential stability of the pose estimation EKF. As seen in Fig. 5, the controller velocities applied to the camera are bounded, and go to zero as the current camera pose approaches the desired pose. In light of

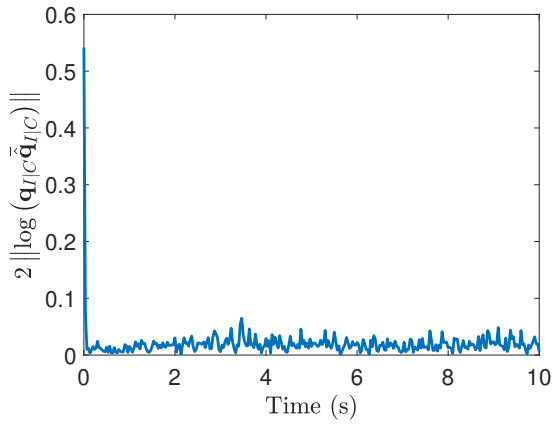

Fig. 6. The logarithmic error between the actual pose and the filter estimate over time.

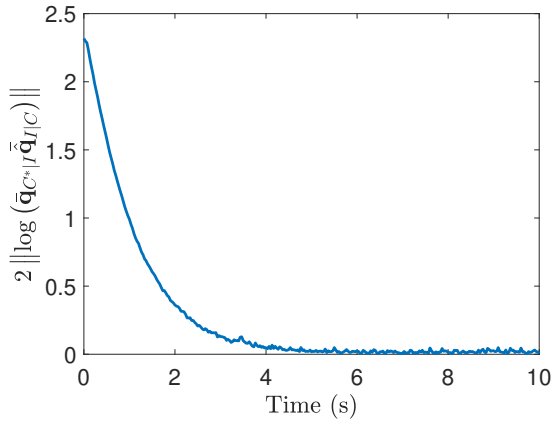

Fig. 7. The logarithmic error between the goal pose and the current pose estimate over time.

Theorem 1, the convergence of the estimation error in Fig. 6 and a stable controller results in the asymptotic stability of the system. This is validated through convergence of the logarithmic error in Fig. 7.

VII. Conclusion

In this paper, a PBVS control method is outlined for the dual quaternion parametrization. A dual quaternion-based EKF is utilized to estimate the relative pose between the camera and object. A stable proportional logarithmic controller is designed using the EKF estimate. The stability of the observer-controller framework is verified using a nonlinear separation principle. Experimental analysis validates the proposed method in a common visual servoing scenario. In future work, experiments will be done on actual robots to validate the performance of the proposed algorithm.

References

[1] F. Chaumette and S. Hutchinson, "Visual servo control. Part I: Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[2] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 507–515, 2001.

[3] A. P. Dani, N. Gans, and W. E. Dixon, "Position-based visual servo control of leader-follower formation using image-based relative pose and relative velocity estimation," in *American Control Conference*, 2009, pp. 5271–5276.

[4] N. R. Gans and S. A. Hutchinson, "Stable visual servoing through hybrid switched-system control," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 530–540, 2007.

[5] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.

[6] É. Marchand and F. Chaumette, "Virtual visual servoing: a framework for real-time augmented reality," in *Computer Graphics Forum*, vol. 21, no. 3. Wiley Online Library, 2002, pp. 289–297.

[7] L. Quan and Z. Lan, "Linear n-point camera pose determination," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 21, no. 8, pp. 774–780, 1999.

[8] A. P. Dani, N. R. Fischer, and W. E. Dixon, "Single camera structure and motion," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 238–243, 2011.

[9] N. Gans, A. Dani, and W. Dixon, "Visual servoing to an arbitrary pose with respect to an object given a single known length," in *2008 American Control Conference*. IEEE, 2008, pp. 1261–1267.

[10] W. J. Wilson, C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.

[11] S. Léonard, E. A. Croft, and J. J. Little, "Dynamic visibility checking for vision-based motion planning," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2283–2288.

[12] O. Tahri, Y. Mezouar, F. Chaumette, and H. Araujo, "Visual servoing and pose estimation with cameras obeying the unified model," in *Visual Servoing via Advanced Numerical Methods*. Springer, 2010, pp. 231–250.

[13] K. Fathian, J. P. Ramirez-Paredes, E. A. Doucette, J. W. Curtis, and N. R. Gans, "Quest: A quaternion-based approach for camera motion estimation from minimal feature points," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 857–864, 2018.

[14] G. Hu, N. Gans, N. Fitz-Coy, and W. Dixon, "Adaptive homography-based visual servo tracking control via a quaternion formulation," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 1, pp. 128–135, 2009.

[15] E. Özgür and Y. Mezouar, "Kinematic modeling and control of a robot arm using unit dual quaternions," *Robotics and Autonomous Systems*, vol. 77, pp. 66–73, 2016.

[16] H. Dong, Q. Hu, M. R. Akella, and F. Mazenc, "Partial lyapunov strictification: Dual-quaternion-based observer for 6-dof tracking control," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2453–2469, 2018.

[17] I. S. Godage and I. D. Walker, "Dual quaternion based modal kinematics for multisection continuum arms," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 1416–1422.

[18] G. Krishnan, J. Bishop-Moser, C. Kim, and S. Kota, "Kinematics of a generalized class of pneumatic artificial muscles," *Journal of Mechanisms and Robotics*, vol. 7, no. 4, 2015.

[19] S. Li, D. M. Vogt, D. Rus, and R. J. Wood, "Fluid-driven origami-inspired artificial muscles," *Proceedings of the National academy of Sciences*, vol. 114, no. 50, pp. 13 132–13 137, 2017.

[20] T. Olsson, J. Bengtsson, A. Robertsson, and R. Johansson, "Visual position tracking using dual quaternions with hand-eye motion constraints," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 3. IEEE, 2003, pp. 3491–3496.

[21] D. Han, Q. Wei, Z. Li, and W. Sun, "Control of oriented mechanical systems: A method based on dual quaternion," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 3836–3841, 2008.

[22] K. Reif, F. Sonnemann, and R. Unbehauen, "An EKF-based nonlinear observer with a prescribed degree of stability," *Automatica*, vol. 34, no. 9, pp. 1119–1123, 1998.

[23] F. Viel, E. Busvelle, and J.-P. Gauthier, "Stability of polymerization reactors using i/o linearization and a high-gain observer," *Automatica*, vol. 31, no. 7, pp. 971–984, 1995.

[24] E. Cervera, "Visual servoing toolbox for matlab / simulink," 2003.