A Quaternion Formulation for Homography-based Visual Servo Control

G. Hu, W. E. Dixon, S. Gupta, and N. Fitz-Coy

Department of Mechanical and Aerospace Engineering,
University of Florida, Gainesville, FL 32611-6250
Email: 

Abstract—Previous homography-based visual servo controllers have been developed using an error system that contains a singularity resulting from the representation of the rotation matrix. For some aerospace applications such as visual servo control of satellites or air vehicles, the singularity introduced by the rotation representation may be restrictive. To eliminate this singularity, a homography-based visual servo controller is developed in this paper based on an error system composed of the unit quaternion representation. The proposed adaptive controller regulates a camera to a desired position and orientation that is determined from a desired image. A quaternion-based Lyapunov function is developed to facilitate the control design and the stability analysis.

I. INTRODUCTION

Visual servo control methods have been developed where the feedback signal is composed of pure image-space information (i.e., image-based visual servo control) [7], [8], reconstructed Euclidean information (i.e., position-based visual servo control) [2], [13], [21], [25] or a blend of both image information and reconstructed information (i.e., 2.5D or homography-based visual servo control) [1], [3]-[5], [9], [10], [17]-[20]. From a review of these approaches (see [15] for an in-depth discussion), one significant advantage of homography-based visual servo controllers is that singularities in the image-Jacobian are avoided (i.e., the image-Jacobian is typically upper triangular and invertible in homography-based approaches). However, all of the previous homography-based visual servo controllers have been developed using an error system that contains a singularity resulting from trigonometric terms in the Euler angle-axis representation of the rotation matrix. In [20], a discussion is provided that explains how the singularity in the error system in Euler angle-axis representation is manifested differently for various task functions.

For some aerospace applications such as visual servo control of satellites or air vehicles, the singularity introduced by the rotation representation may be restrictive. The contribution of this paper is the development of an error system and visual servo controller based on the quaternion formulation that eliminates the singularity in the rotation representation.

A homography can be constructed from image pairs and decomposed via textbook methods (e.g., [11], [14]). Once the rotation matrix has been determined, development is provided that illustrates how an error system can be constructed in terms of the unit quaternion. The resulting rotation error system is void of trigonometric terms contained in previous error systems. A controller is developed and proven to regulate a camera to a desired position and orientation that is determined from a desired image. The controller contains an adaptive feedforward term to compensate for the unknown distance from the camera to the observed planar patch. A quaternion-based Lyapunov function is developed to facilitate the control design and the stability analysis.

II. MODEL FORMULATION

A. Euclidean Relationships

Fig. 1. Coordinate frame relationships between a camera viewing a planar patch at different spatiotemporal instances.

Without loss of generality1, the subsequent development is based on the assumption that an object (e.g., the end-effector of a robot manipulator, the wing of an aircraft, corners of a tumbling satellite) has four coplanar and non-collinear feature points denoted by \( O_i \; \forall i = 1, 2, 3, 4 \). The plane defined by the four feature points is denoted by \( \pi \) as depicted in Fig. 1. The coordinate frame \( \mathcal{F} \) in Fig. 1 is affixed to a camera viewing the object, and the stationary coordinate frame \( \mathcal{F}^* \) denotes a

1If four coplanar target points are not available then the subsequent development can exploit the classic eight-points algorithm [19] with no four of the eight target points being coplanar.
constant desired camera position and orientation that is defined by a desired image. The vectors \( \tilde{m}_i(t), \tilde{m}_i^* \in \mathbb{R}^3 \) in Fig. 1 are defined as

\[
m_i \triangleq \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T
\]

\[
m_i^* \triangleq \begin{bmatrix} x_i^* & y_i^* & z_i^* \end{bmatrix}^T
\]

where \( x_i(t), y_i(t), z_i(t) \in \mathbb{R} \) and \( x_i^*, y_i^*, z_i^* \in \mathbb{R} \) denote the Euclidean coordinates of the feature points \( O_i \) expressed in the frames \( \mathcal{F} \) and \( \mathcal{F}^* \), respectively. From standard Euclidean geometry, relationships between \( \tilde{m}_i(t) \) and \( \tilde{m}_i^* \) can be determined as

\[
m_i = x_f + R \tilde{m}_i^*
\]

where \( x_f(t) \in \mathbb{R}^3 \) denotes the translation vector expressed in \( \mathcal{F} \), and \( R(t) \in SO(3) \) denotes the rotation between \( \mathcal{F} \) and \( \mathcal{F}^* \). As also illustrated in Fig. 1, \( n^* \in \mathbb{R}^3 \) denotes the constant unit normal to the plane \( \pi \), and the constant distance from the origin of \( \mathcal{F}^* \) to \( \pi \) along the unit normal is denoted by \( d^* \triangleq n^T \tilde{m}_i^* \in \mathbb{R} \). The normalized Euclidean coordinates, denoted by \( m_i(t) \in \mathbb{R}^3 \) and \( m_i^* \in \mathbb{R}^3 \), are defined as

\[
m_i \triangleq \frac{\tilde{m}_i}{z_i} = \begin{bmatrix} \frac{x_i}{z_i} & \frac{y_i}{z_i} & 1 \end{bmatrix}^T
\]

\[
m_i^* \triangleq \frac{\tilde{m}_i^*}{z_i^*} = \begin{bmatrix} \frac{x_i^*}{z_i^*} & \frac{y_i^*}{z_i^*} & 1 \end{bmatrix}^T
\]

with the standard assumption that \( z_i(t) > \delta \) and \( z_i^* > \delta \) where \( \delta \) is an arbitrarily small positive constant. From (2) and (3), the relationship in (1) can be expressed as

\[
m_i = \frac{z_i^*}{z_i} \left( R + \frac{x_f(t)}{d^*} n^T \right) m_i^*
\]

where \( \alpha_i(t) \in \mathbb{R} \) is a scaling term, and \( H(t) \in \mathbb{R}^{3 \times 3} \) denotes the Euclidean homography.

**B. Projective Relationships**

Each feature point on \( \pi \) has a projected pixel coordinate \( p_i(t) \in \mathbb{R}^3 \) and \( p_i^* \in \mathbb{R}^3 \) in \( \mathcal{F} \) and \( \mathcal{F}^* \) respectively, denoted by

\[
p_i \triangleq \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T
\]

\[
p_i^* \triangleq \begin{bmatrix} u_i^* & v_i^* & 1 \end{bmatrix}^T
\]

where \( u_i(t), v_i(t), u_i^*, v_i^* \in \mathbb{R} \). The projected pixel coordinates \( p_i(t) \) and \( p_i^* \) are related to the normalized task-space coordinates \( m_i(t) \) and \( m_i^* \) by the following global invertible transformation (i.e., the pinhole camera model)

\[
p_i = Am_i \quad p_i^* = Am_i^*
\]

where \( A \in \mathbb{R}^{3 \times 3} \) is a constant, upper triangular, and invertible intrinsic camera calibration matrix that is explicitly defined as [14]

\[
A \triangleq \begin{bmatrix}
\alpha & -\alpha \cot \phi & u_0 \\
0 & \frac{\beta}{\sin \phi} & v_0 \\
0 & 0 & 1
\end{bmatrix}
\]

In (8), \( u_0, v_0 \in \mathbb{R} \) denote the pixel coordinates of the principal point (i.e., the image center that is defined as the frame buffer coordinates of the intersection of the optical axis with the image plane), \( \alpha, \beta \in \mathbb{R} \) represent the product of the camera scaling factors and the focal length, and \( \phi \in \mathbb{R} \) is the skew angle between the camera axes. Based on (7), the Euclidean relationship in (4) can be expressed in terms on the image coordinates as

\[
p_i = \alpha_i AHA^{-1} p_i^*.
\]

A set of 12 linearly independent equations given by the 4 feature point pairs \( (p_i^*, p_i(t)) \) with 3 independent equations per feature point can be developed from (9) to determine the projective homography up to a scalar multiple (i.e., the product \( \alpha_i AHA^{-1} \) can be determined). Various methods can then be applied (e.g., see [12], [26]) to decompose the Euclidean homography, to obtain \( \alpha_i(t), H(t), R(t), \frac{x_f(t)}{d^*}, \text{and } n^* \). The rotation matrix \( R(t) \) and the depth ratio \( \alpha_i(t) \) are used in the subsequent control design.

**C. Unit Quaternion Representation of the Rotation Matrix**

One of the outcomes of the homography decomposition is a computed rotation matrix between \( \mathcal{F} \) and \( \mathcal{F}^* \). From this rotation matrix, several different representations can be utilized to develop the error system. In previous homography-based visual servo control literature, the Euler angle-axis representation has been used to describe the rotation matrix. The angle-axis parameters \( (\varphi, k) \), where \( \varphi(t) \in \mathbb{R} \) represents a rotation angle about a suitable unit vector \( k(t) \in \mathbb{R}^3 \), can be easily calculated (e.g., using the algorithm shown in [24]) using the unit quaternion parameterization. This parameterization facilitates the subsequent problem formulation, control development, and stability analysis since the unit quaternion provides a global nonsingular parameterization of the rotation matrix between \( \mathcal{F} \) and \( \mathcal{F}^* \). Given \( (\varphi, k) \), the unit quaternion vector

\[
q \triangleq \begin{bmatrix} q_0 & q_o^T \end{bmatrix}^T
\]

can be constructed as

\[
\begin{bmatrix}
q_0(t) \\
q_o(t)
\end{bmatrix} = \begin{bmatrix}
\cos \left( \frac{\varphi(t)}{2} \right) \\
\frac{\varphi(t)}{2} \cos \left( \frac{\varphi(t)}{2} \right)
\end{bmatrix}
\]

where \( q_o(t) \triangleq \begin{bmatrix} q_1(t) & q_2(t) & q_3(t) \end{bmatrix}^T, q_i(t) \in \mathbb{R} \forall i = 0, ..., 3 \), and the following nonlinear constraint must be satisfied

\[
q^T q = 1.
\]

Given unit vector \( k(t) \) and angle \( \varphi(t) \), the rotation matrix \( R(t) = e^{k \times \varphi} \) can be calculated using the Rodrigues formula [16]

\[
R = e^{k \times \varphi} = I_3 + k^\times \sin(\varphi) + (k^\times)^2(1 - \cos(\varphi))
\]

where \( I_3 \) is the \( 3 \times 3 \) identity matrix, and the notation \( k^\times(t) \)
denotes the following skew-symmetric form of the vector $k(t)$:

$$
k^\times = \begin{bmatrix}
0 & -k_3 & k_2 \\
k_3 & 0 & -k_1 \\
-k_2 & k_1 & 0
\end{bmatrix}; \quad \forall k = [k_1 \ k_2 \ k_3]^T. \tag{14}
$$

The rotation matrix in (13) can be expressed as

$$
R(q) = \begin{bmatrix}
q_0^2 - q_4^2 q_v & 2q_4 q_v^2 + 2q_0 q_v^x \\
2(q_4 q_v + q_4 q_0) & q_0^2 - q_1^2 q_v - q_2^2 q_v + 2q_3 q_v^x \\
2(q_4 q_v + q_4 q_0) & 2(q_4 q_v - q_5 q_0) \\
2(q_4 q_v - q_5 q_0) & q_0^2 - q_3^2 q_v - q_2^2 q_v + 2q_3 q_v^x
\end{bmatrix} \tag{15}
$$

$$
= I_3 + 2q_0 q_v^x + 2(q_v^x)^2
$$

$$
\begin{bmatrix}
q_0^2 - q_1^2 q_v - q_2^2 q_v + 2q_3 q_v^x \\
2(q_4 q_v + q_4 q_0) \\
2(q_4 q_v - q_5 q_0) \\
q_0^2 - q_3^2 q_v - q_2^2 q_v + 2q_3 q_v^x
\end{bmatrix}
$$

after utilizing (11). From (15) various approaches could be used to determine $q_0(t)$ and $q_v(t)$; however, numerical significance of the resulting computations can be lost if $q_0(t)$ is close to zero [23]. In [23], a method is developed to determine $q_0(t)$ and $q_v(t)$ that provides robustness against such computational issues. Specifically, the diagonal terms of $R(q)$ can be obtained from (15) as

$$
R_{11} = 1 + 2(-q_2^2 - q_3^2) \tag{16}
$$

$$
R_{22} = 1 + 2(-q_1^2 - q_3^2) \tag{17}
$$

$$
R_{33} = 1 + 2(-q_1^2 - q_2^2) \tag{18}
$$

By utilizing (12) and (16)-(18), the following expressions can be developed [23]:

$$
q_0^2 = \frac{R_{11} + R_{22} + R_{33} + 1}{4} \tag{19}
$$

$$
q_{v1}^2 = \frac{R_{11} - R_{22} - R_{33} + 1}{4} \tag{20}
$$

$$
q_{v2}^2 = \frac{R_{22} - R_{11} - R_{33} + 1}{4} \tag{21}
$$

$$
q_{v3}^2 = \frac{R_{33} - R_{11} - R_{22} + 1}{4} \tag{22}
$$

where $q_0(t)$ is restricted to be non-negative without loss of generality (this restriction enables the minimum rotation to be obtained). As stated in [23], the greatest numerical accuracy for computing $q_0(t)$ and $q_v(t)$ is obtained by using the element in (19) with the largest value and then computing the remaining terms respectively. For example, if $q_0^2(t)$ has the maximum value in (19) then the greatest numerical accuracy can be obtained by computing $q_0(t)$ and $q_v(t)$ as

$$
q_0 = \sqrt{\frac{R_{11} + R_{22} + R_{33} + 1}{4}}
$$

$$
q_{v1} = \frac{R_{32} - R_{23}}{4q_0}
$$

$$
q_{v2} = \frac{R_{21} - R_{12}}{4q_0}
$$

$$
q_{v3} = \frac{R_{13} - R_{31}}{4q_0}
$$

Likewise, if $q_{v1}^2(t)$ has the maximum value in (19) then the greatest numerical accuracy can be obtained by computing $q_0(t)$ and $q_v(t)$ as

$$
q_0 = \frac{R_{32} - R_{23}}{4q_{v1}}
$$

$$
q_{v2} = \pm \sqrt{\frac{R_{11} - R_{22} - R_{33} + 1}{4}}
$$

$$
q_{v3} = \frac{R_{12} + R_{21}}{4q_{v2}}
$$

where the sign of $q_{v3}(t)$ is selected so that $q_0(t) \geq 0$. If $q_{v2}(t)$ is the maximum, then

$$
q_0 = \frac{R_{13} - R_{31}}{4q_{v2}}
$$

$$
q_{v1} = \frac{R_{12} - R_{21}}{4q_{v1}}
$$

$$
q_{v3} = \frac{R_{23} + R_{32}}{4q_{v3}}
$$

or if $q_{v3}(t)$ is the maximum, then

$$
q_0 = \frac{R_{21} - R_{12}}{4q_{v3}}
$$

$$
q_{v1} = \frac{R_{4v3} + R_{4v3}}{4q_{v2}}
$$

$$
q_{v2} = \frac{R_{23} + R_{32}}{4q_{v3}}
$$

where the sign of $q_{v2}(t)$ or $q_{v3}(t)$ is selected so that $q_0(t) \geq 0$.

The results from (20)-(23) are that given the rotation matrix $R(t)$ from the homography decomposition, the unit quaternion vector can be determined that represents the rotation without introducing a singularity as in previous visual servo control literature. The expressions in (20)-(23) will be utilized in the subsequent control development and stability analysis.

Remark 1: The results in (20)-(23) differ from the algorithm presented in [23] because of a difference in the definition of the rotation matrix. Specifically, in comparison to the rotation matrix in (15), the rotation matrix defined in [23] is

$$
R(q) = (q_0^2 - q_v^T q_v) I_3 + 2q_v q_v^T - 2q_0 q_v^x. \tag{24}
$$

The difference is because the rotation matrix in the current paper relates the moving coordinate frame $F$ to the fixed coordinate frame $F^*$ (similar to robotics literature where the coordinate frame attached to the moving end-effector is related to the base frame), whereas the rotation matrix in [23] relates the fixed coordinate frame to the moving coordinate frame (as is typical in aerospace literature).

### III. CONTROL OBJECTIVE

The control objective is to regulate a camera to a desired position and orientation. This objective is based on the assumption that the linear and angular velocities of the camera...
are control inputs that can be independently controlled (i.e., unconstrained motion) and that the camera is calibrated (i.e., the parameters in the $A$ matrix in (8) are known). In the Euclidean-space (see Fig. 1), this objective can be quantified as

$$R(t) \rightarrow I_3 \quad \text{as} \quad t \rightarrow \infty$$  \hspace{1cm} (25)$$

and

$$\|x_f(t)\| \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty. \hspace{1cm} (26)$$

Based on (25), the rotation regulation objective can also be quantified as the desire to regulate $q_c(t)$ in the sense that

$$\|q_c(t)\| \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty. \hspace{1cm} (27)$$

From the structure of (15), if the objective in (27) is satisfied then (25) is also satisfied. To construct the translation error system, we define the extended normalized coordinates, denoted by $m_e(t) \in \mathbb{R}^3$ and $m_e^* \in \mathbb{R}^3$, as defined in [9], [18]

$$m_e = \begin{bmatrix} m_{e1} & m_{e2} & m_{e3} \end{bmatrix}^T$$

$$m_e^* = \begin{bmatrix} x_i & y_i & z_i \\ x_i^* & y_i^* & z_i^* \end{bmatrix}^T \ln z_i.$$

The difference in the extended normalized coordinates is defined by the translation regulation error, denoted by $e(t) \in \mathbb{R}^3$, as follows:

$$e = m_e - m_e^*$$

$$= \begin{bmatrix} x_i - x_i^* & y_i - y_i^* & z_i - z_i^* \end{bmatrix}^T \ln z_i \hspace{1cm} (30)$$

Based on (30), the translation regulation objective can be quantified as the desire to regulate $e(t)$ in the sense that

$$\|e(t)\| \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty. \hspace{1cm} (31)$$

From the structure of (30), if the objective in (31) is satisfied then

$$m_i(t) \rightarrow m_i^* \quad \text{and} \quad z_i(t) \rightarrow z_i^* \quad \text{as} \quad t \rightarrow \infty. \hspace{1cm} (32)$$

If (25) and (32) are satisfied, then (4) can be used to conclude that (26) is also satisfied. The subsequent section will target the control development based on the objectives in (27) and (31).

IV. CONTROL DEVELOPMENT

A. Open-Loop Error System

The open-loop rotation error system can be developed by taking the time derivative of $q(t)$ as [6]

$$\dot{q} = B(q)\omega_c \hspace{1cm} (33)$$

where $\omega_c(t) \in \mathbb{R}^3$ denotes the angular velocity of the camera with respect to $\mathcal{F}^*$, and $B(q) \in \mathbb{R}^{4 \times 3}$ is defined as

$$B(q) = \frac{1}{2} \begin{bmatrix} -q_v^T \\ q_0 I_3 - q_v \times \end{bmatrix}. \hspace{1cm} (34)$$

The open-loop translation error system can be derived in the same manner as in [4] and [19] as

$$\dot{z}_v^* = -\alpha_t L_v v_c + L_w \omega_c z_v^* \hspace{1cm} (35)$$

where $v_c(t) \in \mathbb{R}^3$ denotes the linear velocity of the camera with respect to $\mathcal{F}^*$, and the Jacobian-like matrices $L_v(t)$, $L_w(t) \in \mathbb{R}^{3 \times 3}$ are defined as

$$L_v = \begin{bmatrix} 1 & 0 & -m_{e1} \\ 0 & 1 & -m_{e2} \\ 0 & 0 & 1 \end{bmatrix} \hspace{1cm} (36)$$

$$L_w = \begin{bmatrix} m_{e1} m_{e2} & -1 - m_{e1} m_{e2} & m_{e1} \\ 1 + m_{e2}^2 & -m_{e1} m_{e2} & -m_{e1} \\ -m_{e2} & m_{e1} & 0 \end{bmatrix}. \hspace{1cm} (37)$$

B. Closed-Loop Error System

Based on the open-loop error system in (33) and the subsequent stability analysis, the angular velocity controller is designed as

$$\omega_c = -K_{\omega}(I_3 - q_v^*)^{-1} q_v \hspace{1cm} (38)$$

where $K_{\omega} \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains. The controller in (38) does not exhibit a singularity since

$$\text{det}(I_3 - q_v^*) = 1 + \|q_v\|^2 \neq 0. \hspace{1cm} (39)$$

After substituting (38) into (33), the following closed-loop error rotation system can be developed

$$\dot{q}_0 = \frac{1}{2} q_v^T K_{\omega}(I_3 - q_v^*)^{-1} q_v \hspace{1cm} (40)$$

$$\dot{q}_v = -\frac{1}{2} K_{\omega} (q_0 I_3 - q_v^*) (I_3 - q_v^*)^{-1} q_v. \hspace{1cm} (41)$$

The translational control input $v_c(t)$ is designed as

$$v_c = \frac{1}{\alpha_t} L_v^{-1}(K_v e + z_v^* L_w \omega_c) \hspace{1cm} (42)$$

where $K_v \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains, and $z_v^*(t) \in \mathbb{R}$ represents an estimate for the unknown constant $z_v^*$ that is defined as follows:

$$\dot{z}_v^* = e^T L_w \omega_c. \hspace{1cm} (43)$$

The controller in (42) does not exhibit a singularity since the determinant of $L_v(t)$ is invertible and $\alpha_t(t) > 0$. After substituting (42) into (35) the following closed-loop translation error system can be obtained:

$$\dot{z}_v^* = -K_v e + \ddot{z}_i L_w \omega_c \hspace{1cm} (44)$$

where $\dddot{z}_i(t) \in \mathbb{R}$ denotes the following parameter estimation
error:
\[ \dot{z}_i = z_i^* - \hat{z}_i^*. \]  

(45)

The closed-loop estimation error is determined by taking the time derivative of (45) and utilizing (43) as
\[ \dot{\hat{z}}_i = -e^T \dot{L}_w \omega_c. \]  

(46)

C. Stability Analysis

Theorem 1: The controller given in (38) and (42) along with the adaptive update law in (43) ensures global asymptotic translation and rotation regulation in the sense that
\[ \| q_v(t) \| \to 0 \quad \text{and} \quad \| e(t) \| \to 0 \quad \text{as} \quad t \to \infty. \]  

(47)

Proof: Let \( V(t) \in \mathbb{R} \) denote the following non-negative positive definite function (i.e., a Lyapunov candidate):
\[ V = \frac{z^*_i}{2} e^T e + q_v^T q_v \]  

(48)

The time-derivative of \( V(t) \) can be determined as
\[ \dot{V} = e^T (-K_v e + \dot{L}_w \omega_c) \]  

(49)

\[ + q_v^T K_w (q_v I_3 - q_v^*) \left( I_3 - q_v^* \right)^{-1} q_v \]  

\[ - (1 - q_v^*) q_v^T K_w (I_3 - q_v^*)^{-1} q_v \]  

\[ - \dot{\hat{z}}_i e^T \dot{L}_w \omega_c \]  

where (40), (41), (44), and (46) were utilized. The following negative semi-definite expression is obtained after simplifying (49):
\[ \dot{V} = -e^T K_v e - q_v^T K_w q_v. \]  

(50)

Based on (48) and (50), \( e(t), q_v(t), q_0(t), \) \( \dot{z}_i(t) \in \mathcal{L}_2 \) and \( e(t), q_v(t) \in \mathcal{L}_2 \). Since \( \dot{z}_i(t) \in \mathcal{L}_\infty \), it is clear from (45) that \( \dot{z}_i^*(t) \in \mathcal{L}_\infty \). Based on the fact that \( e(t), q_v(t), q_0(t) \in \mathcal{L}_\infty \), we can utilize (28), (30), (36), (37) to prove that \( m_c(t), L_v(t), L_w(t) \in \mathcal{L}_\infty \). Based on the fact that \( \dot{z}_i^*(t), e(t), q_v(t), q_0(t), L_w(t) \in \mathcal{L}_\infty \), we can utilize (38) and (42) to prove that \( v_c(t), \omega_c(t) \in \mathcal{L}_\infty \). From the previous results, (33)-(35) can be used to prove that \( \dot{e}(t), \dot{q}_v(t) \in \mathcal{L}_\infty \). Since \( e(t), q_v(t) \in \mathcal{L}_\infty \cap \mathcal{L}_2 \) and \( \dot{e}(t), \dot{q}_v(t) \in \mathcal{L}_\infty \), we can utilize a corollary to Barbalat’s Lemma [22] to conclude the result given in (47).

VI. C ONCLUSIONS

Two adaptive visual servo controllers are presented that achieve asymptotic regulation of the camera translation and rotation error systems. The contribution of this paper is that the presented controller is formulated based on a quaternion representation of a rotation matrix that is computed from a homography decomposition. By developing the error systems and controller based on a homography decomposition, singularities associated with the typical image-Jacobian are eliminated. By utilizing the quaternion formulation, singularities associated with other representations of the rotation matrix used in previous literature are also eliminated. A position-based rotation and translation controller is proven to achieve the regulation result through a Lyapunov-based stability analysis. An extension is provided that also explains how a controller with both image-space information and reconstructed Euclidean information could be used to achieve the same stability result. Future efforts will target experimental demonstration of the developed controller and extensions for tracking a desired time-varying trajectory as described in [4].

REFERENCES


