

Adaptive Homography-Based Visual Servo Tracking for a Fixed Camera Configuration With a Camera-in-Hand Extension

Jian Chen, Darren M. Dawson, Warren E. Dixon, and Aman Behal

Abstract—In this brief, a homography-based adaptive visual servo controller is developed to enable a robot end-effector to track a desired Euclidean trajectory as determined by a sequence of images for both the camera-in-hand and fixed-camera configurations. To achieve the objectives, a Lyapunov-based adaptive control strategy is employed to actively compensate for the lack of unknown depth measurements and the lack of an object model. The error systems are constructed as a hybrid of pixel information and reconstructed Euclidean variables obtained by comparing the images and decomposing a homographic relationship. Simulation results are provided to demonstrate the performance of the developed controller for the fixed camera configuration.

Index Terms—Adaptive control, control applications, Lyapunov-methods, visual servo control.

I. INTRODUCTION

A KEY issue that impacts camera-based visual servo control is the relationship between the Euclidean-space and the image-space. One factor that impacts this relationship is the fact that the image-space is a two-dimensional (2-D) projection of the three-dimensional (3-D) Euclidean-space. To compensate for the lack of depth information from the 2-D image data, some researchers have focused on the use of alternate sensors (e.g., laser and sound ranging technologies). Other researchers have explored the use of a camera-based vision system in conjunction with other sensors along with sensor fusion methods or the use of additional cameras in a stereo configuration that triangulate on corresponding images. However, the practical drawbacks of incorporating additional sensors include: increased cost, increased complexity, decreased reliability, and increased processing burden. Motivated by these practical constraints, recent research [2], [5], [10]–[12], [17] has focused on monocular camera-based visual servo strategies that rely on analytic techniques to address the lack of depth information. One strategy that has recently been employed involves the use of partitioning methods that exploit a combination of reconstructed 3-D Euclidean information and 2-D image-space information. For example, in the series of papers

by Malis and Chaumette [1], [2], [18], [19] various kinematic control strategies exploit the fact that the interaction between translation and rotation components can be decoupled through a homography (i.e., a geometric transformation). Specifically, information combined from the reconstructed Euclidean-space and the image-space is utilized to regulate the translation and rotation error systems. In [8], Deguchi utilizes a homography relationship and an epipolar condition to decouple the rotation and translation components and then illustrates how two types of visual controllers can be developed from the decoupled information. Corke and Hutchinson [5] also developed a hybrid image-based visual servoing scheme that decouples rotation and translation components from the remaining degrees of freedom. One drawback of some of the aforementioned controllers are claims (without a supporting proof) that a constant, best-guess estimate of the depth information can be utilized in lieu of the exact value. Motivated by the desire to actively compensate for unmeasurable depth information, Conticelli developed an adaptive kinematic controller in [3] to ensure uniformly ultimately bounded (UUB) set-point regulation, provided conditions on the translational velocity and bounds on the uncertain depth parameters are satisfied. In [4], Conticelli *et al.* proposed a 3-D depth estimation procedure that exploits a prediction error provided a positive definite condition on the interaction matrix is satisfied. In [10] and [11], Fang *et al.* recently developed homography-based visual servo controllers to asymptotically regulate a manipulator end-effector and a mobile robot, respectively, by developing an adaptive update law that actively compensates for an unknown depth parameter. In [12], Fang *et al.* also developed a camera-in-hand regulation controller that incorporated a robust control structure to compensate for uncertainty in the extrinsic calibration parameters.

After examining the literature, it is evident that much of the previous visual servo controllers have only been designed to address the regulation problem. That is, the objective of most of the previous controllers is to force a hand-held camera to a Euclidean position defined by a static reference image. Unfortunately, many practical applications require a robotic system to move along a predefined or dynamically changing trajectory. For example, a human operator may predefine an image trajectory through a high-level interface, and this trajectory may need to be modified on-the-fly to respond to obstacles moving in and out of the environment. Moreover, it is well known that a regulating controller may produce erratic behavior and require excessive initial control torques if the initial error is large. Motivated by the need for new advancements to meet visual servo tracking applications, previous research has concentrated on developing different types of path planning techniques in the image-space [6], [21]–[23]. More recently, Mezouar and Chaumette developed a path-fol-

Manuscript received May 21, 2003; revised June 29, 2004. Manuscript received in final form April 8, 2005. Recommended by Associate Editor M. Jankovic. This research was supported in part by Air Force Office of Scientific Research under Contract F49620-03-1-0381 at the University of Florida and in part by two DOC grants, an ARO Automotive Center grant, a DOE contract, a Honda Corporation grant, and a DARPA contract at Clemson University.

J. Chen and D. M. Dawson are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA.

W. E. Dixon is with the Mechanical and Aerospace Engineering Department, University of Florida, Gainesville, FL 32611 USA.

A. Behal is with the Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY 13699 USA (e-mail: wdixon@ufl.edu).

Digital Object Identifier 10.1109/TCST.2005.852150

lowing image-based visual servo algorithm in [20] where the path to a goal point is generated via a potential function that incorporates motion constraints. In [7], Cowan *et al.* develop a hybrid position/image-space controller that forces a manipulator to a desired setpoint while avoiding obstacles and ensuring the object remains in the field-of-view by avoiding pitfalls such as self-occlusion.

In contrast to the approaches in [7] and [20] in which a path is planned as a means to reach a desired setpoint, hybrid tracking controllers are developed in this brief where the robot end-effector is required to track a prerecorded time-varying reference trajectory. To develop the hybrid controllers, a homography-based visual servoing approach is utilized. The motivation for using this approach is that the visual servo control problem can be incorporated with a Lyapunov-based control design strategy to overcome many practical and theoretical obstacles associated with more traditional, purely image-based approaches. Specifically, one of the challenges of this problem is that the translation error system is corrupted by an unknown depth-related parameter. By formulating a Lyapunov-based argument, an adaptive update law is developed to actively compensate for the unknown depth parameter. In addition, the proposed approach facilitates: 1) translation/rotational control in the full six degree-of-freedom task-space without the requirement of an object model, 2) partial servoing on pixel data that yields improved robustness and increases the likelihood that the centroid of the object remains in the camera field-of-view [19], and 3) the use of an image Jacobian that is only singular for multiples of 2π , in contrast to the state-dependent singularities present in the image Jacobians associated with many of the purely image-based controllers. The homography-based controllers in this brief target both the fixed camera and the camera-in-hand configurations. The control development for the fixed camera problem is presented in detail, and the camera-in-hand problem is included as an extension.

This brief is organized in the following manner. In Section II, a geometric model is constructed that relates the Euclidean coordinates of the target points. In Section III, a Euclidean homography is developed that relates the normalized Euclidean coordinates of the target points, and the pinhole camera model is used to develop a projective homography that relates the measurable pixel coordinates. In Sections IV and V, the control objective is developed along with the open-loop and closed-loop error dynamics. The stability of the closed-loop error system is analyzed through Lyapunov-based methods in Section VI. An extension to the camera-in-hand problem is presented in Section VII. Simulation results illustrating the performance of the control law are given in Section VIII, and concluding remarks are given in Section IX.

II. GEOMETRIC MODEL

To make the subsequent development more tractable, four target points located on an object (i.e., the end-effector of a robot manipulator) denoted by $O_i \forall i = 1, 2, 3, 4$ are considered to be coplanar¹ and not colinear. Based on this assumption, consider a

¹It should be noted that if four coplanar target points are not available then the subsequent development can exploit the classic eight-points algorithm [18] with no four of the eight target points being coplanar.

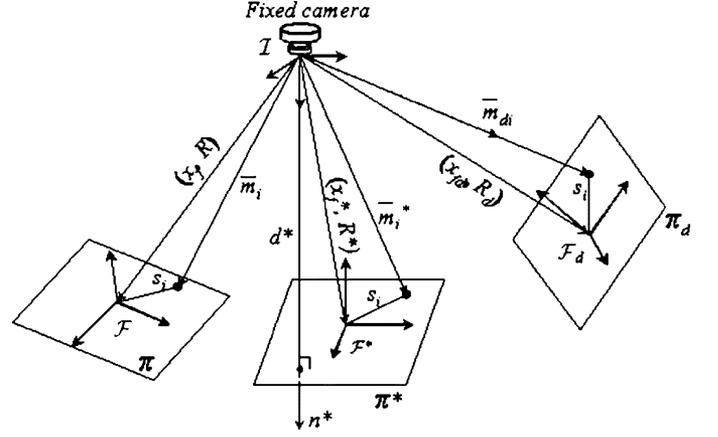


Fig. 1. Coordinate frame relationships between a fixed camera and the plane defined by the current, desired, and reference feature points (i.e., π , π_d , and π^*).

fixed plane, denoted by π^* , that is defined by a reference image of the object. In addition, consider the actual and desired motion of the plane containing the end-effector target points, denoted by π and π_d , respectively (see Fig. 1). To develop a relationship between the planes, an inertial coordinate system, denoted by \mathcal{I} , is defined where the origin coincides with the center of a fixed camera. The Euclidean coordinates of the target points on π , π_d , and π^* can be expressed in terms of \mathcal{I} , respectively, as follows:

$$\begin{aligned} \bar{m}_i(t) &\triangleq [x_i(t) \quad y_i(t) \quad z_i(t)]^T \\ \bar{m}_{di}(t) &\triangleq [x_{di}(t) \quad y_{di}(t) \quad z_{di}(t)]^T \\ \bar{m}_i^* &\triangleq [x_i^* \quad y_i^* \quad z_i^*]^T \end{aligned} \quad (1)$$

under the standard assumption that the distances from the origin of \mathcal{I} to the target points remains positive (i.e., $z_i(t)$, $z_{di}(t)$, $z_i^* > \varepsilon$ where ε denotes an arbitrarily small positive constant). Orthogonal coordinate systems \mathcal{F} , \mathcal{F}_d , and \mathcal{F}^* are attached to the planes π , π_d , and π^* , respectively (see Fig. 1). To relate the coordinate systems, let $R(t)$, $R_d(t)$, $R^* \in SO(3)$ denote the rotation between \mathcal{F} and \mathcal{I} , \mathcal{F}_d and \mathcal{I} , and \mathcal{F}^* and \mathcal{I} , respectively, and let $x_f(t)$, $x_{fd}(t)$, $x_f^* \in \mathbb{R}^3$ denote the respective translation vectors expressed in the coordinates of \mathcal{I} . As also illustrated in Fig. 1, $n^* \in \mathbb{R}^3$ denotes the constant unit normal to the plane π^* expressed in the coordinates of \mathcal{I} , and $s_i \in \mathbb{R}^3$ denotes the constant coordinates of the i -th target point. The constant distance from the origin of \mathcal{I} to π^* along the unit normal is denoted by $d^* \in \mathbb{R}$ and is defined as follows:

$$d^* \triangleq n^{*T} \bar{m}_i^*. \quad (2)$$

From the geometry between the coordinate frames depicted in Fig. 1, the following relationships can be developed:

$$\begin{aligned} \bar{m}_i &= x_f + R s_i \\ \bar{m}_{di} &= x_{fd} + R_d s_i \\ \bar{m}_i^* &= x_f^* + R^* s_i. \end{aligned} \quad (3)$$

After solving the third equation in (3) for s_i and then substituting the resulting expression into the first and second equations, the following relationships can be obtained:

$$\bar{m}_i = \bar{x}_f + \bar{R} \bar{m}_i^* \quad \bar{m}_{di} = \bar{x}_{fd} + \bar{R}_d \bar{m}_i^* \quad (4)$$

where $\bar{R}(t), \bar{R}_d(t) \in SO(3)$ and $\bar{x}_f(t), \bar{x}_{fd}(t) \in \mathbb{R}^3$ are new rotational and translational variables, respectively, defined as follows (see the Appendix for further insight into the geometrical significance of the new rotational and translational variables):

$$\begin{aligned} \bar{R} &= R(R^*)^T & \bar{R}_d &= R_d(R^*)^T \\ \bar{x}_f &= x_f - \bar{R}x_f^* & \bar{x}_{fd} &= x_{fd} - \bar{R}_d x_{fd}^*. \end{aligned} \quad (5)$$

From (2), the relationships in (4) can be expressed as follows:

$$\bar{m}_i = \left(\bar{R} + \frac{\bar{x}_f}{d^*} n^{*T} \right) \bar{m}_i^* \quad \bar{m}_{di} = \left(\bar{R}_d + \frac{\bar{x}_{fd}}{d^*} n^{*T} \right) \bar{m}_{di}^*. \quad (6)$$

Remark 1: The subsequent development requires that the constant rotation matrix R^* be known. The constant rotation matrix R^* can be obtained *a priori* using various methods (e.g., a second camera, Euclidean measurements). The subsequent development is also based on the assumption that the target points do not become occluded.

III. EUCLIDEAN RECONSTRUCTION

The relationship given by (6) provides a means to quantify a translation and rotation error between \mathcal{F} and \mathcal{F}^* and between \mathcal{F}_d and \mathcal{F}^* . Since the Euclidean position of \mathcal{F} , \mathcal{F}_d , and \mathcal{F}^* cannot be directly measured, a Euclidean reconstruction is developed in this section to obtain the position and rotational error information by comparing multiple images acquired from the fixed, monocular vision system. Specifically, comparisons are made between the current image, the reference image obtained *a priori*, and the *a priori* known sequence of images that define the trajectory of \mathcal{F}_d . To facilitate the subsequent development, the normalized Euclidean coordinates of the points on π , π_d , and π^* can be, respectively, expressed in terms of \mathcal{I} as $m_i(t)$, $m_{di}(t)$, $m_i^* \in \mathbb{R}^3$, as follows:

$$m_i \triangleq \frac{\bar{m}_i}{z_i} = \begin{bmatrix} x_i & y_i & 1 \\ z_i & z_i & 1 \end{bmatrix}^T \quad (7)$$

$$m_{di} \triangleq \frac{\bar{m}_{di}}{z_{di}} = \begin{bmatrix} x_{di} & y_{di} & 1 \\ z_{di} & z_{di} & 1 \end{bmatrix}^T \quad (8)$$

$$m_i^* \triangleq \frac{\bar{m}_i^*}{z_i^*} = \begin{bmatrix} x_i^* & y_i^* & 1 \\ z_i^* & z_i^* & 1 \end{bmatrix}^T. \quad (9)$$

From the expressions given in (6)–(9), the rotation and translation between the coordinate systems can now be related in terms of the normalized coordinates as follows:

$$m_i = \underbrace{\frac{z_i^*}{z_i}}_{\alpha_i} \underbrace{\left(\bar{R} + \frac{\bar{x}_f n^{*T}}{d^*} \right)}_H m_i^* \quad (10)$$

$$m_{di} = \underbrace{\frac{z_{di}^*}{z_{di}}}_{\alpha_{di}} \underbrace{\left(\bar{R}_d + \frac{\bar{x}_{fd} n^{*T}}{d^*} \right)}_{H_d} m_{di}^* \quad (11)$$

where $\alpha_i(t), \alpha_{di}(t) \in \mathbb{R}$ denote invertible depth ratios, $H(t), H_d(t) \in \mathbb{R}^{3 \times 3}$ denote Euclidean homographies [13], and $\bar{x}_h(t),$

$\bar{x}_{hd}(t) \in \mathbb{R}^3$ denote scaled translation vectors that are defined as follows:

$$\bar{x}_h = \frac{\bar{x}_f}{d^*} \quad \bar{x}_{hd} = \frac{\bar{x}_{fd}}{d^*}. \quad (12)$$

Each target point on π , π_d , and π^* will have a projected pixel coordinate expressed in terms of \mathcal{I} , denoted by $u_i(t), v_i(t) \in \mathbb{R}$ for π , $u_{di}(t), v_{di}(t) \in \mathbb{R}$ for π_d , and $u_i^*, v_i^* \in \mathbb{R}$ for π^* , that are defined as follows:

$$\begin{aligned} p_i &\triangleq [u_i \quad v_i \quad 1]^T & p_{di} &\triangleq [u_{di} \quad v_{di} \quad 1]^T \\ p_i^* &\triangleq [u_i^* \quad v_i^* \quad 1]^T. \end{aligned} \quad (13)$$

In (13), $p_i(t), p_{di}(t), p_i^* \in \mathbb{R}^3$ represent the image-space coordinates of the time-varying target points, the desired time-varying target point trajectory, and the constant reference target points, respectively. To calculate the Euclidean homography given in (10) and (11) from pixel information, the projected pixel coordinates of the target points are related to $m_i(t), m_{di}(t)$, and m_i^* by the following pin-hole lens models [13]:

$$p_i = A m_i \quad p_{di} = A m_{di} \quad p_i^* = A m_i^* \quad (14)$$

where $A \in \mathbb{R}^{3 \times 3}$ is a known, constant, and invertible intrinsic camera calibration matrix. After substituting (14) into (10) and (11), the following relationships can be developed:

$$p_i = \alpha_i \underbrace{(A H A^{-1})}_G p_i^* \quad p_{di} = \alpha_{di} \underbrace{(A H_d A^{-1})}_{G_d} p_{di}^* \quad (15)$$

where $G(t) = [g_{ij}(t)]$, $G_d(t) = [g_{dij}(t)] \forall i, j = 1, 2, 3 \in \mathbb{R}^{3 \times 3}$ denote projective homographies. From the first relationship in (15), a set of 12 linearly independent equations given by the four target point pairs $(p_i^*, p_i(t))$ with three independent equations per target pair can be used to determine the projective homography up to a scalar multiple (i.e., the product $\alpha_i(t)G(t)$ can be determined). From the definition of $G(t)$ given in (15), various techniques can then be used [14], [26] to decompose the Euclidean homography, to obtain $\alpha_i(t), G(t), H(t)$, and the rotation and translation signals $\bar{R}(t)$ and $\bar{x}_h(t)$, and n^* . Likewise, by using the target point pairs $(p_{di}^*, p_{di}(t))$, the desired Euclidean homography can be decomposed to obtain $\alpha_{di}(t), G_d(t), H_d(t)$, and the desired rotation and translation signals $\bar{R}_d(t)$ and $\bar{x}_{hd}(t)$. The rotation matrices $R(t)$ and $R_d(t)$ can be computed from $\bar{R}(t)$ and $\bar{R}_d(t)$ by using (5) and the fact that R^* is assumed to be known. Hence, $R(t), \bar{R}(t), R_d(t), \bar{R}_d(t), \bar{x}_h(t), \bar{x}_{hd}(t)$, and the depth ratios $\alpha_i(t)$ and $\alpha_{di}(t)$ are all known signals that can be used for control synthesis.

IV. CONTROL OBJECTIVE

The objective is to develop a visual servo controller that ensures that the trajectory of \mathcal{F} tracks \mathcal{F}_d (i.e., $\bar{m}_i(t)$ tracks $\bar{m}_{di}(t)$), where the trajectory of \mathcal{F}_d is constructed relative to the reference camera position/orientation given by \mathcal{F}^* . To ensure that $\bar{m}_i(t)$ tracks $\bar{m}_{di}(t)$ from the Euclidean reconstruction given in (10) and (11), the tracking control objective can be

stated as follows:² $\bar{R}(t) \rightarrow \bar{R}_d(t)$, $m_1(t) \rightarrow m_{d1}(t)$, and $z_1(t) \rightarrow z_{d1}(t)$ (and, hence, $\bar{x}_h(t) \rightarrow \bar{x}_{hd}(t)$). The 3-D control objective is complicated by the fact that only 2-D image information is measurable. That is, while the development of the homography provides a means to reconstruct some Euclidean information, the formulation of a controller is challenging due to the fact that the time varying signals $z_1(t)$ and $z_{d1}(t)$ are not measurable. In addition, it is desirable to servo on actual pixel information (in lieu of reconstructed Euclidean information) to improve robustness to intrinsic camera calibration parameters and to increase the likelihood that the object will stay in the field of view of the camera [19].

To reformulate the control objective in light of these issues, a hybrid translation tracking error, denoted by $e_v(t) \in \mathbb{R}^3$, is defined as follows:

$$e_v = p_e - p_{ed} \quad (16)$$

where $p_e(t), p_{ed}(t) \in \mathbb{R}^3$ are defined as follows:

$$\begin{aligned} p_e &= [u_1 \quad v_1 \quad -\ln(\alpha_1)]^T \\ p_{ed} &= [u_{d1} \quad v_{d1} \quad -\ln(\alpha_{d1})]^T \end{aligned} \quad (17)$$

and $\ln(\cdot)$ denotes the natural logarithm. A rotation tracking error, denoted by $e_\omega(t) \in \mathbb{R}^3$, is defined as follows:

$$e_\omega \triangleq \Theta - \Theta_d \quad (18)$$

where $\Theta(t), \Theta_d(t) \in \mathbb{R}^3$ denote the axis-angle representation of $\bar{R}(t)$ and $\bar{R}_d(t)$ as follows [25]:

$$\Theta = u(t)\theta(t) \quad \Theta_d = u_d(t)\theta_d(t). \quad (19)$$

For the representations in (19), $u(t), u_d(t) \in \mathbb{R}^3$ represent unit rotation axes, and $\theta(t), \theta_d(t) \in \mathbb{R}$ denote the respective rotation angles about $u(t)$ and $u_d(t)$ that are assumed to be confined to the following regions:

$$-\pi < \theta(t) < \pi \quad -\pi < \theta_d(t) < \pi. \quad (20)$$

Based on the error system formulations in (16) and (18), the control objective can be stated as the desire to regulate the tracking error signals $e_v(t)$ and $e_\omega(t)$ to zero. If the tracking error signals $e_v(t)$ and $e_\omega(t)$ are regulated to zero then the object can be proven to be tracking the desired trajectory (see the Appendix for further details).

Remark 2: A particular solution for $\theta(t)$ and $u(t)$ can be determined as follows [25]:

$$\theta = \cos^{-1} \left(\frac{1}{2} (\text{tr}(\bar{R}) - 1) \right) \quad [u]_\times = \frac{\bar{R} - \bar{R}^T}{2 \sin(\theta)} \quad (21)$$

where the notation $\text{tr}(\cdot)$ denotes the trace of a matrix, and $[u]_\times$ denotes the 3×3 skew-symmetric expansion of $u(t)$.

Remark 3: To develop a tracking control design, it is typical that the desired trajectory is used as a feedforward component

²Any point O_i can be utilized in the subsequent development. However, to reduce the notational complexity, we have elected to select the image point O_1 and, hence, the subscript 1 is utilized in lieu of i in the subsequent development.

in the control design. Hence, for a kinematic controller the desired trajectory is required to be at least first-order differentiable and at least second-order differentiable for a dynamic level controller. To this end, a sufficiently smooth function (e.g., a spline function) is used to fit the sequence of target points to generate the desired trajectory $p_{di}(t)$. Hence, it is assumed that $p_{ed}(t)$ and $\dot{p}_{ed}(t)$ are bounded functions of time. From the projective homography introduced in (15), $p_{di}(t)$ can be expressed in terms of the *a priori* known, functions $\alpha_{di}(t)$, $H_d(t)$, $\bar{R}_d(t)$, and $\bar{x}_{hd}(t)$. Since these signals can be obtained from the prerecorded sequence of images, sufficiently smooth functions can also be generated for these signals by fitting a sufficiently smooth spline function to the signals. Hence, in practice, the *a priori* developed smooth functions $\alpha_{di}(t)$, $\bar{R}_d(t)$, and $\bar{x}_{hd}(t)$ can be constructed as bounded functions with bounded time derivatives. Based on the assumption that $\bar{R}_d(t)$ is a bounded first-order differentiable function with a bounded derivative, (21) can be used to conclude that $u_d(t)$ and $\theta_d(t)$ are bounded first-order differentiable functions with a bounded derivative. Hence, $\Theta_d(t)$ and $\dot{\Theta}_d(t)$ can be assumed to be bounded. In the subsequent tracking control development, the desired signals $\dot{p}_{ed}(t)$ and $\dot{\Theta}_d(t)$ will be used as a feedforward control term.

V. CONTROL FORMULATION

A. Open-Loop Error System

To develop the open-loop error system for $e_\omega(t)$, we take the time derivative of (18) to obtain the following expression (see the Appendix for further details):

$$\dot{e}_\omega = L_\omega R \omega_e - \dot{\Theta}_d. \quad (22)$$

In (22), the Jacobian-like matrix $L_\omega(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$L_\omega = I_3 - \frac{\theta}{2} [u]_\times + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\frac{\theta}{2})} \right) [u]_\times^2 \quad (23)$$

where

$$\text{sinc}(\theta(t)) \triangleq \frac{\sin \theta(t)}{\theta(t)}$$

and $\omega_e(t) \in \mathbb{R}^3$ denotes the angular velocity of the object expressed in \mathcal{F} . By exploiting the fact that $u(t)$ is a unit vector (i.e., $\|u\|^2 = 1$), the determinant of $L_\omega(t)$ can be calculated as [17]

$$\det(L_\omega) = \frac{1}{\text{sinc}^2(\frac{\theta}{2})} \quad (24)$$

where $\det(\cdot)$ signifies the determinant operator. From (24), it is clear that $L_\omega(t)$ is only singular for multiples of 2π (i.e., out of the assumed workspace); therefore, $L_\omega(t)$ is invertible in the assumed workspace. To develop the open-loop error system for $e_v(t)$, we take the time derivative of (16) to obtain the following expression (see the Appendix for further details):

$$z_1^* \dot{e}_v = \alpha_1 A_e L_v R [v_e + [\omega_e]_\times s_1] - z_1^* \dot{p}_{ed} \quad (25)$$

where $v_e(t) \in \mathbb{R}^3$ denotes the linear velocity of the object expressed in \mathcal{F} . In (25), $A_e \in \mathbb{R}^{3 \times 3}$ is defined as follows:

$$A_e = A - \begin{bmatrix} 0 & 0 & u_0 \\ 0 & 0 & v_0 \\ 0 & 0 & 0 \end{bmatrix} \quad (26)$$

where $u_0, v_0 \in \mathbb{R}$ denote the pixel coordinates of the principal point,³ and the auxiliary Jacobian-like matrix $L_v(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$L_v = \begin{bmatrix} 1 & 0 & -\frac{x_1}{z_1} \\ 0 & 1 & -\frac{y_1}{z_1} \\ 0 & 0 & 1 \end{bmatrix}. \quad (27)$$

Remark 4: It is easy to show that the product $A_e L_v$ is an invertible upper triangular matrix from (26) and (27).

B. Closed-Loop Error System

Based on the structure of the open-loop error systems and subsequent stability analysis, the angular and linear camera velocity control inputs for the object are defined as follows:

$$\omega_e = R^T L_\omega^{-1} (\dot{\Theta}_d - K_\omega e_\omega) \quad (28)$$

$$v_e = -\frac{1}{\alpha_1} R^T (A_e L_v)^{-1} (K_v e_v - \hat{z}_1^* \dot{p}_{ed}) - [\omega_e]_\times \hat{s}_1. \quad (29)$$

In (28) and (29), $K_\omega, K_v \in \mathbb{R}^{3 \times 3}$ denote diagonal matrices of positive constant control gains, and $\hat{z}_1^*(t) \in \mathbb{R}$, $\hat{s}_1(t) \in \mathbb{R}^3$ denote parameter estimates that are generated according to the following adaptive update laws:

$$\dot{\hat{z}}_1^*(t) = -\gamma_1 e_v^T \dot{p}_{ed} \quad (30)$$

$$\dot{\hat{s}}_1 = -\alpha_1 \Gamma_2 [\omega_e]_\times R^T L_v^T A_e^T e_v \quad (31)$$

where $\gamma_1 \in \mathbb{R}$ denotes a positive constant adaptation gain, and $\Gamma_2 \in \mathbb{R}^{3 \times 3}$ denotes a positive constant diagonal adaptation gain matrix. After substituting (28) into (22), the following closed-loop error dynamics can be obtained:

$$\dot{e}_\omega = -K_\omega e_\omega. \quad (32)$$

After substituting (29) into (25), the closed-loop translation error dynamics can be determined as follows:

$$z_1^* \dot{e}_v = -K_v e_v + \alpha_1 A_e L_v R [\omega_e]_\times \tilde{s}_1 - \tilde{z}_1^* \dot{p}_{ed} \quad (33)$$

where the parameter estimation error signals $\tilde{z}_1^*(t) \in \mathbb{R}$ and $\tilde{s}_1(t) \in \mathbb{R}^3$ are defined as follows:

$$\tilde{z}_1^* = z_1^* - \hat{z}_1^* \quad \tilde{s}_1 = s_1 - \hat{s}_1. \quad (34)$$

From (32) it is clear that the angular velocity control input given in (28) is designed to yield an exponentially stable rotational error system. The linear velocity control input given in (29) and the adaptive update laws given in (30) and (31) are motivated to yield a negative feedback term in translational error system with additional terms included to cancel out cross-product terms involving the parameter estimation errors in the subsequent stability analysis.

³The principal point is the image center that is defined as the frame buffer coordinates of the intersection of the optical axis with the image plane.

VI. STABILITY ANALYSIS

Theorem 1: The control inputs designed in (28) and (29), along with the adaptive update laws defined in (30) and (31), ensure that $e_\omega(t)$ and $e_v(t)$ are asymptotically driven to zero in the sense that

$$\lim_{t \rightarrow \infty} \|e_\omega(t)\|, \|e_v(t)\| = 0. \quad (35)$$

Proof: To prove Theorem 1, a nonnegative function $V(t) \in \mathbb{R}$ is defined as follows:

$$V \triangleq \frac{1}{2} e_\omega^T e_\omega + \frac{z_1^*}{2} e_v^T e_v + \frac{1}{2\gamma_1} \tilde{z}_1^{*2} + \frac{1}{2} \tilde{s}_1^T \Gamma_2^{-1} \tilde{s}_1. \quad (36)$$

After taking the time derivative of (36) and then substituting for the closed-loop error systems developed in (32) and (33), the following expression can be obtained:

$$\begin{aligned} \dot{V} = & e_v^T (-K_v e_v + \alpha_1 A_e L_v R [\omega_e]_\times \tilde{s}_1 - \tilde{z}_1^* \dot{p}_{ed}) \\ & - e_\omega^T K_\omega e_\omega - \frac{1}{\gamma_1} \tilde{z}_1^* \dot{\tilde{z}}_1^* - \tilde{s}_1^T \Gamma_2^{-1} \dot{\tilde{s}}_1 \end{aligned} \quad (37)$$

where the time derivative of (34) was utilized. After substituting the adaptive update laws designed in (30) and (31) into (37), the following simplified expression can be obtained:

$$\dot{V} = -e_\omega^T K_\omega e_\omega - e_v^T K_v e_v \quad (38)$$

where the fact that $[\omega_e]_\times^T = -[\omega_e]_\times$ was utilized. Based on (34), (36), and (38), it can be determined that $e_\omega(t)$, $e_v(t)$, $\tilde{z}_1^*(t)$, $\hat{z}_1^*(t)$, $\tilde{s}_1(t)$, $\hat{s}_1(t) \in \mathcal{L}_\infty$ and that $e_\omega(t)$, $e_v(t) \in \mathcal{L}_2$ [9]. Based on the assumption that $\dot{\Theta}_d(t)$ is designed as a bounded function, the expressions given in (18), (23), (24), and (28) can be used to conclude that $\omega_e(t) \in \mathcal{L}_\infty$. Since $e_v(t) \in \mathcal{L}_\infty$, (7), (13), (14), (16), (17), and (27) can be used to prove that $m_1(t)$, $L_v(t) \in \mathcal{L}_\infty$. Given that $\dot{p}_{ed}(t)$ is assumed to be bounded function, the expressions in (29)–(33) can be used to conclude that $\dot{\tilde{z}}_1^*(t)$, $\dot{\hat{s}}_1(t)$, $v_e(t)$, $\dot{e}_v(t)$, $\dot{e}_\omega(t) \in \mathcal{L}_\infty$. Since $e_\omega(t)$, $e_v(t) \in \mathcal{L}_2$ and $e_\omega(t)$, $\dot{e}_\omega(t)$, $e_v(t)$, $\dot{e}_v(t) \in \mathcal{L}_\infty$, Barbalat's Lemma [24] can be used to prove the result given in (35). \square

Remark 5: The result in (35) is practically global in the sense that it is valid over the entire domain with the exception of the singularity introduced by the exponential parameterization of the rotation matrix (20) and the physical restriction that $z_i(t)$, $z_i^*(t)$, and $z_{di}(t)$ must remain positive. Although the result stated in Theorem 5 indicates asymptotic convergence for the rotation error $e_\omega(t)$, it is evident from (32) that

$$e_\omega(t) \leq e_\omega(0) \exp(-\lambda_{\min}(K_\omega)t)$$

where $\lambda_{\min}(K_\omega)$ denotes the minimum eigenvalue of the constant matrix K_ω . However, the fact that $e_\omega(t) \leq e_\omega(0) \exp(-\lambda_{\min}(K_\omega)t)$ does not simplify the control development or stability analysis and the overall resulting control objective of tracking a desired set of prerecorded images is still asymptotically achieved.

VII. CAMERA-IN-HAND EXTENSION

Based on the development provided for the fixed camera problem in the previous sections, a controller for the

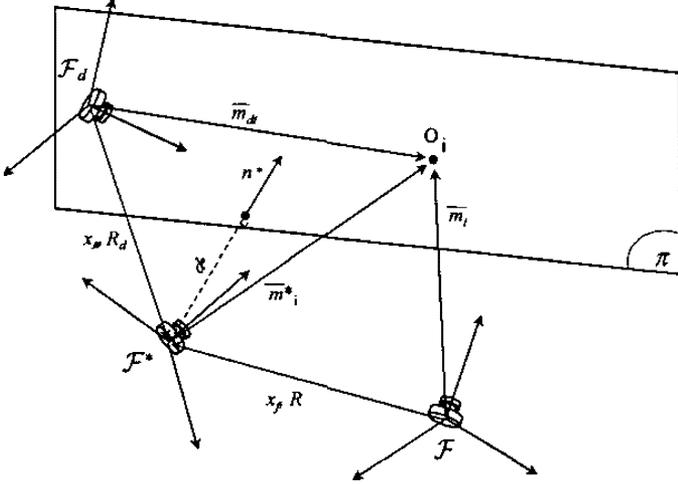


Fig. 2. Coordinate frame relationships between the fixed feature point plane and the camera-in-hand at the current, desired, and reference position and orientation (i.e., \mathcal{F} , \mathcal{F}_d , and \mathcal{F}^*).

camera-in-hand problem can be developed in a similar manner. To formulate a controller for the camera-in-hand tracking problem consider the geometric relationships depicted in Fig. 2 where the camera is held by a robot end-effector (not shown). The coordinate frames \mathcal{F} , \mathcal{F}_d , and \mathcal{F}^* depicted in Fig. 2 are attached to the camera and denote the actual, desired, and reference locations for the camera, respectively. From the geometry between the coordinate frames, \bar{m}_i^* can be related to $\bar{m}_i(t)$ and $\bar{m}_{di}(t)$ as follows:

$$\begin{aligned}\bar{m}_i &= x_f + R\bar{m}_i^* \\ \bar{m}_{di} &= x_{fd} + R_d\bar{m}_i^*\end{aligned}\quad (39)$$

where $\bar{m}_i(t)$, $\bar{m}_{di}(t)$, and \bar{m}_i^* now denote the Euclidean coordinates of O_i expressed in \mathcal{F} , \mathcal{F}_d , and \mathcal{F}^* , respectively. In (39), $R(t)$, $R_d(t) \in SO(3)$ denote the rotation between \mathcal{F} and \mathcal{F}^* and between \mathcal{F}_d and \mathcal{F}^* , respectively, and $x_f(t)$, $x_{fd}(t) \in \mathbb{R}^3$ denote translation vectors from \mathcal{F} to \mathcal{F}^* and \mathcal{F}_d to \mathcal{F}^* expressed in the coordinates of \mathcal{F} and \mathcal{F}_d , respectively. By utilizing (2), (7)–(9), and a relationship similar to (12), the expressions in (39) can be written as follows:

$$m_i = \alpha_i \underbrace{(R + x_h n^{*T})}_H m_i^* \quad (40)$$

$$m_{di} = \alpha_{di} \underbrace{(R_d + x_{hd} n^{*T})}_H m_i^*. \quad (41)$$

In (40) and (41), $x_h(t)$, $x_{hd}(t) \in \mathbb{R}^3$ denote the following scaled translation vectors:

$$x_h = \frac{x_f}{d^*} \quad x_{hd} = \frac{x_{fd}}{d^*} \quad (42)$$

$\alpha_i(t)$ and $\alpha_{di}(t)$ are introduced in (10) and (11), and $m_i(t)$, $m_{di}(t)$, and m_i^* now denote the normalized Euclidean coordinates of O_i expressed in \mathcal{F} , \mathcal{F}_d , and \mathcal{F}^* , respectively. Based on the development in (39)–(41), the Euclidean reconstruction and control formulation can be developed in the same manner as for the fixed camera problem. Specifically, the signals $R(t)$, $R_d(t)$, $x_h(t)$, $x_{hd}(t)$, and the depth ratios $\alpha_i(t)$ and $\alpha_{di}(t)$ can be computed. The error systems for the camera-in-hand problem are de-

veloped the same as for the fixed camera problem (16)–(19). However, $u(t)$, $u_d(t)$, $\theta(t)$, and $\theta_d(t)$ are defined as in (21) in terms of $R(t)$ and $R_d(t)$, respectively, for the camera-in-hand problem. Based on this fact, the open-loop error dynamics for the rotation system can be derived as follows:

$$\dot{e}_\omega = -L_\omega \omega_c - \dot{\Theta}_d \quad (43)$$

where the fact that

$$[\omega_c]_\times = -\dot{R}R^T \quad (44)$$

is used, and $\omega_c(t)$ denotes the camera angular velocity expressed in \mathcal{F} . After taking the time derivative of (39), the following expression for $\dot{\bar{m}}_1(t)$ can be derived for the camera-in-hand [10]:

$$\dot{\bar{m}}_1 = -v_c + [\bar{m}_1]_\times \omega_c \quad (45)$$

where $v_c(t)$ denotes the linear velocity of the camera expressed in terms of \mathcal{F} . After utilizing (45), the open-loop dynamics for $e_v(t)$ can be determined as follows:

$$z_1^* \dot{e}_v = -\alpha_1 A_e L_v v_c + (A_e L_v [m_1]_\times \omega_c - \dot{p}_{ed}) z_1^* \quad (46)$$

where $e_v(t)$, $p_e(t)$, $p_{ed}(t)$ are defined in (16) and (17).

Based on the open-loop error systems in (43) and (46), the following control signals are designed:

$$\omega_c \triangleq L_\omega^{-1} (K_\omega e_\omega - \dot{\Theta}_d) \quad (47)$$

$$v_c \triangleq \frac{1}{\alpha_1} (A_e L_v)^{-1} (K_v e_v - \dot{z}_1^* p_{ed}) + \frac{1}{\alpha_1} [m_1]_\times \omega_c z_1^* \quad (48)$$

$$\dot{z}_1^* \triangleq \gamma_1 e_v^T (A_e L_v [m_1]_\times \omega_c - \dot{p}_{ed}) \quad (49)$$

resulting in the following closed-loop error systems:

$$\dot{e}_\omega = -K_\omega e_\omega \quad (50)$$

$$z_1^* \dot{e}_v = -K_v e_v + (A_e L_v [m_1]_\times \omega_c - \dot{p}_{ed}) z_1^*. \quad (51)$$

The result in (35) can now be proven for the camera-in-hand problem using the same analysis techniques and the same non-negative function as defined in (36) with the term containing $\dot{z}_1(t)$ eliminated.

VIII. SIMULATION RESULTS

Simulation studies were performed to illustrate the performance of the controller given in (28)–(31). For the simulation, the intrinsic camera calibration matrix is given as follows:

$$A = \begin{bmatrix} f k_u & -f k_u \cot \phi & u_0 \\ 0 & \frac{f k_v}{\sin \phi} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (52)$$

where $u_0 = [257 \text{ pixels}]$, $v_0 = [253 \text{ pixels}]$, $k_u = [101.4 \text{ pixels} \cdot \text{mm}^{-1}]$ and $k_v = [101.4 \text{ pixels} \cdot \text{mm}^{-1}]$ represent camera scaling factors, $\phi = 90$ [Deg] is the angle between the camera axes, and $f = [12.5 \text{ mm}]$ denotes the camera focal length. The control objective is defined in terms of tracking a desired image sequence. For the simulation, the desired image sequence was required to be artificially generated. To generate an artificial image sequence for the simulation, the Euclidean coordinates of four target points were defined as follows:

$$\begin{aligned} s_1 &= [0.1 \quad -0.1 \quad 0]^T & s_2 &= [0.1 \quad 0.1 \quad 0]^T \\ s_3 &= [-0.1 \quad 0.1 \quad 0]^T & s_4 &= [-0.1 \quad -0.1 \quad 0]^T \end{aligned} \quad (53)$$

and the initial translation and rotation between the current, desired, and reference image feature planes were defined as follows:

$$x_f(0) = \begin{bmatrix} -0.3 \\ -0.1 \\ 3.7 \end{bmatrix} \quad x_{fd}(0) = \begin{bmatrix} 0.2 \\ 0.1 \\ 4 \end{bmatrix} \quad x_f^* = \begin{bmatrix} 0.2 \\ 0.1 \\ 4 \end{bmatrix} \quad (54)$$

$$R(0) = \begin{bmatrix} -0.4698 & -0.8660 & -0.1710 \\ -0.6477 & 0.4698 & -0.5997 \\ 0.5997 & -0.1710 & -0.7817 \end{bmatrix} \quad (55)$$

$$R_d(0) = \begin{bmatrix} 0.9568 & -0.2555 & -0.1386 \\ -0.2700 & -0.9578 & -0.0984 \\ -0.1077 & 0.1316 & -0.9854 \end{bmatrix} \quad (56)$$

$$R^* = \begin{bmatrix} 0.9865 & 0.0872 & -0.1386 \\ 0.0738 & -0.9924 & -0.0984 \\ -0.1462 & 0.0868 & -0.9854 \end{bmatrix}. \quad (57)$$

Based on (52)–(57), the initial pixel coordinates can be computed as follows:

$$\begin{aligned} p_1(0) &= [170 \quad 182 \quad 1]^T & p_2(0) &= [110 \quad 213 \quad 1]^T \\ p_3(0) &= [138 \quad 257 \quad 1]^T & p_4(0) &= [199 \quad 224 \quad 1]^T \\ p_{d1}(0) &= [359 \quad 307 \quad 1]^T & p_{d2}(0) &= [343 \quad 246 \quad 1]^T \\ p_{d3}(0) &= [282 \quad 263 \quad 1]^T & p_{d4}(0) &= [298 \quad 324 \quad 1]^T \\ p_1^* &= [349 \quad 319 \quad 1]^T & p_2^* &= [355 \quad 256 \quad 1]^T \\ p_3^* &= [292 \quad 251 \quad 1]^T & p_4^* &= [286 \quad 314 \quad 1]^T. \end{aligned}$$

The time-varying desired image trajectory was then generated by the kinematics of the target plane where the desired linear and angular velocity were selected as follows:

$$\begin{aligned} v_{ed}(t) &= [0.2 \sin(t) \quad 0.3 \sin(t) \quad 0] \text{ [m/s]} \\ \omega_{ed}(t) &= [0 \quad 0 \quad 0.52 \sin(t)] \text{ [rad/s]}. \end{aligned} \quad (58)$$

The desired translational trajectory is given in Fig. 3, and the desired rotational trajectory is depicted in Fig. 4. The generated desired image trajectory is a continuous function. However, in practice the image trajectory would be discretely represented by a sequence of prerecorded images and would require a data interpolation scheme (i.e., a spline function) as described in Remark 3. Hence, a spline function (i.e., the MATLAB spline routine) was utilized to generate a continuous curve to fit the desired image trajectory. For the top two subplots in Fig. 3, the pixel values obtained from the prerecorded image sequence are denoted by an asterisk (only select data points were included for clarity of illustration), and a cubic spline interpolation that was used to fit the data points is illustrated by a solid line. For the bottom subplot in Fig. 3 and all the subplots in Fig. 4, a plus sign denotes reconstructed Euclidean values computed using the prerecorded pixel data, and the spline function is illustrated by a solid line.

The control gains K_v and K_ω and the adaptation gains γ_1 and Γ_2 were adjusted through trial and error to the following values:

$$\begin{aligned} K_v &= \text{diag}\{6, 8, 5\} & K_\omega &= \text{diag}\{0.6, 0.8, 0.7\} \\ \gamma_1 &= 3 \times 10^{-6} & \Gamma_2 &= 10^{-5} \times \text{diag}\{4.2, 5.6, 2.8\}. \end{aligned} \quad (59)$$

The resulting errors between the actual relative translational and rotational of the target with respect to the reference target and

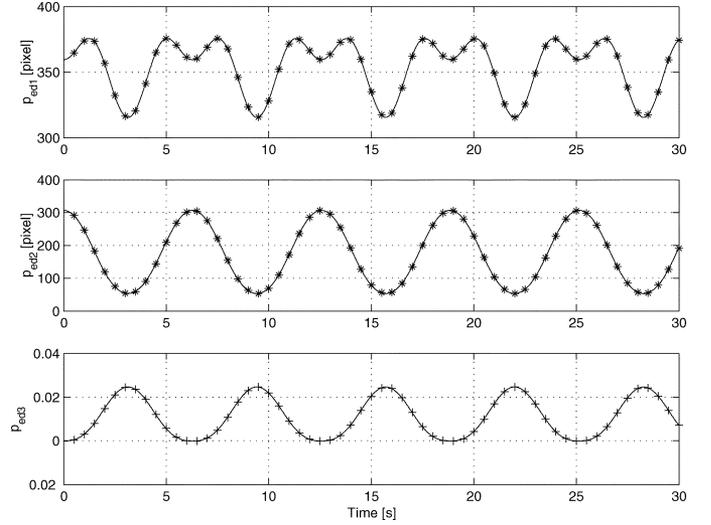


Fig. 3. Desired translational trajectory of the manipulator end-effector generated by a spline function to fit prerecorded image data.

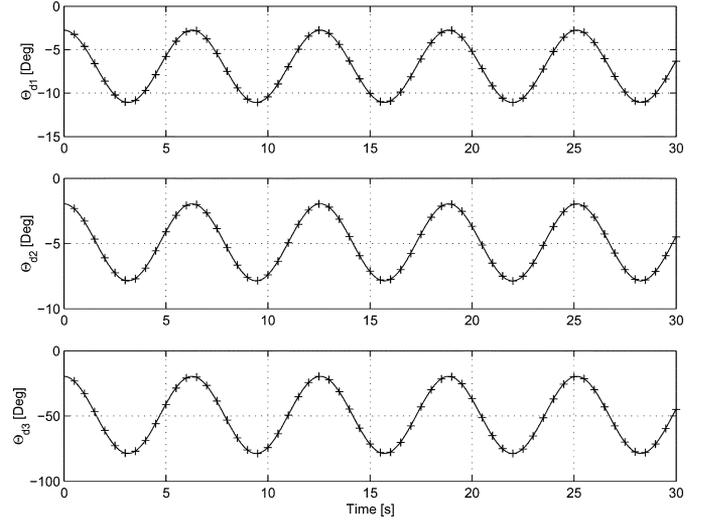


Fig. 4. Desired rotational trajectory of the manipulator end-effector generated by a spline function to fit prerecorded image data.

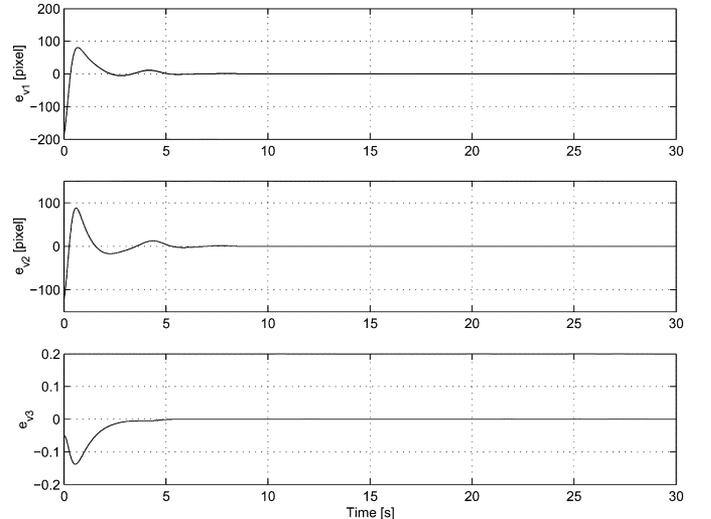


Fig. 5. Error between the actual translation trajectory and the desired translation trajectory given in Fig. 3 for the noise-free case.

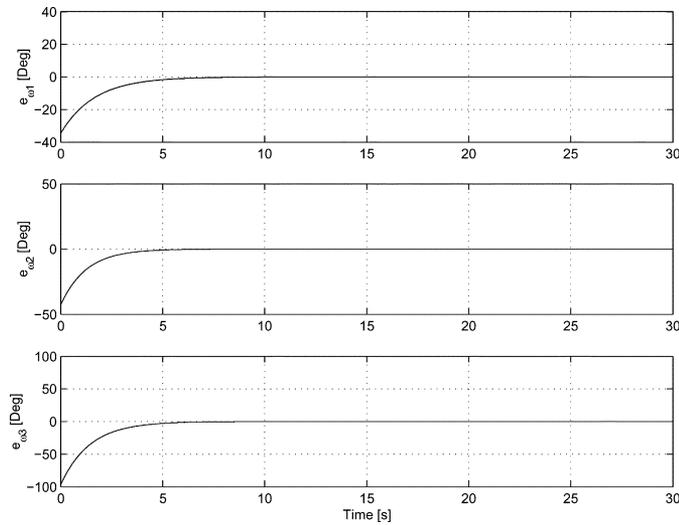


Fig. 6. Error between the actual rotation trajectory and the desired rotation trajectory given in Fig. 4 for the noise-free case.

the desired translational and rotational of the target with respect to the reference target are depicted in Figs. 5 and 6, respectively. The parameter estimate signals are depicted in Figs. 7 and 8. The angular and linear control input velocities (i.e., $\omega_e(t)$ and $v_e(t)$) defined in (28) and (29) are depicted in Figs. 9 and 10.

While the results in Figs. 5–10 provide an example of the performance of the tracking controller under ideal conditions, several issues must be considered for a practical implementation. For example, the performance of the tracking control algorithm is influenced by the accuracy of the image-space feedback signals and the accuracy of the reconstructed Euclidean information obtained from constructing and decomposing the homography. That is, inaccuracies in determining the location of a feature from one frame to the next frame (i.e., feature tracking) will lead to errors in the construction and decomposition of the homography matrix, leading to errors in the feedback control signal. Inaccuracies in determining the feature point coordinates in an image is a similar problem faced in numerous sensor based feedback applications (e.g., noise associated with a force/torque sensor). Practically, errors related to sensor inaccuracies can often be addressed with an ad hoc filter scheme or other mechanisms (e.g., an intelligent image processing and feature tracking algorithm, redundant feature points and an optimal homography computation algorithm).

In light of these practical issues, another simulation was performed where random noise was injected with a standard deviation of 1 pixel (i.e., the measured feature coordinate was subject to ± 4 pixels of measurement error) as in [18]. As in any practical feedback control application in the presence of sensor noise, a filter was employed. Specifically, ad hoc third-order butterworth low-pass filters with a cutoff frequency of 10 rad/s were utilized to preprocess the corrupted image data. The control gains K_v and K_ω and the adaptation gains γ_1 and Γ_2 were tuned through trial and error to the following values:

$$K_v = \text{diag}\{17, 11, 9\} \quad K_\omega = \text{diag}\{0.4, 0.4, 0.4\}$$

$$\gamma_1 = 5 \times 10^{-7} \quad \Gamma_2 = 10^{-5} \times \text{diag}\{2.4, 3.2, 1.6\}. \quad (60)$$

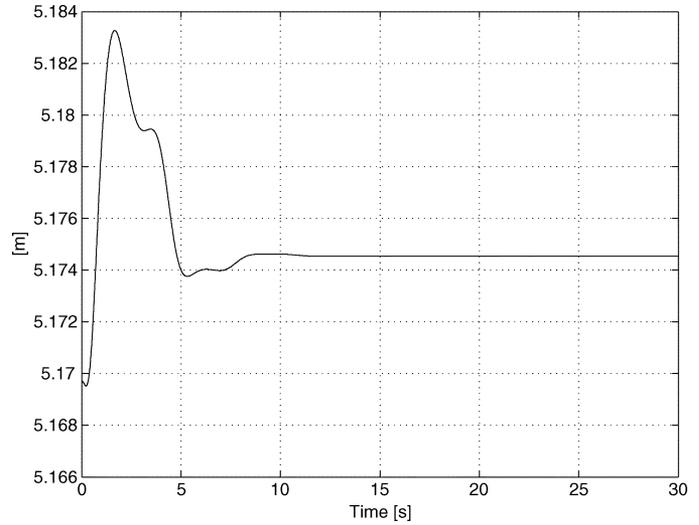


Fig. 7. Parameter estimate for z_1^* for the noise-free case.

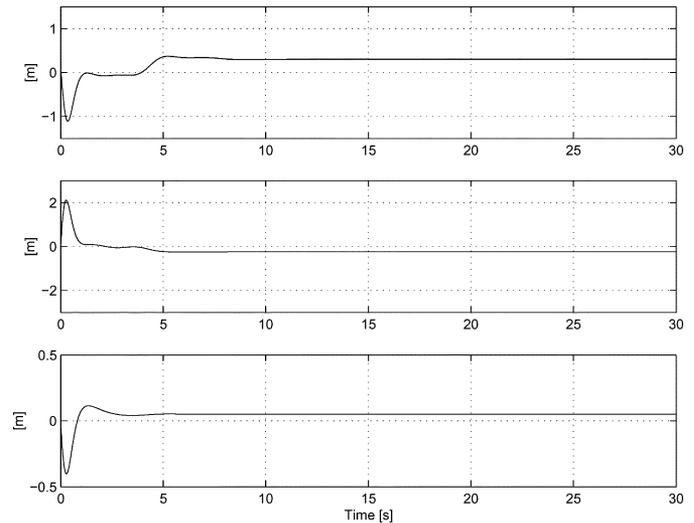


Fig. 8. Parameter estimates for s_1 for the noise-free case.

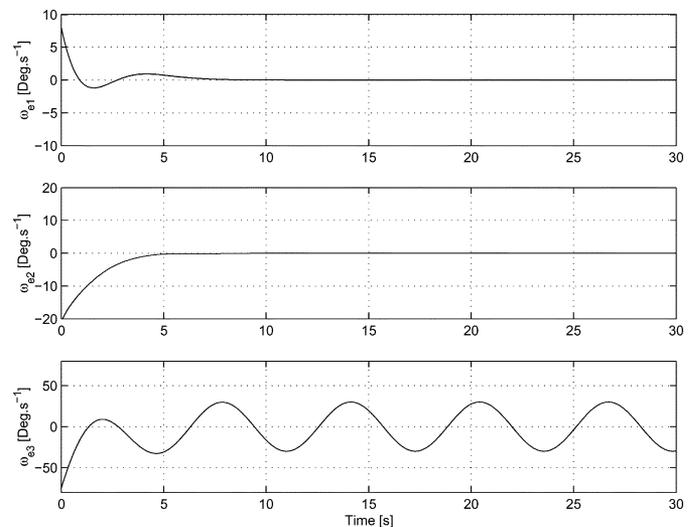


Fig. 9. Angular velocity control input for the noise-free case.

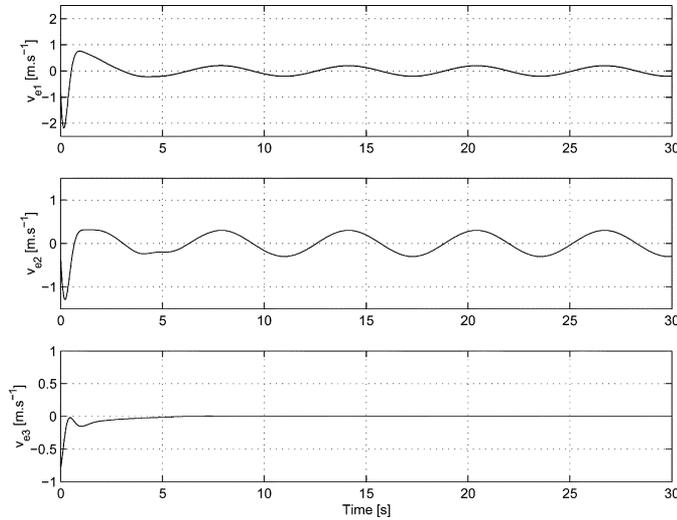


Fig. 10. Linear velocity control input for the noise-free case.

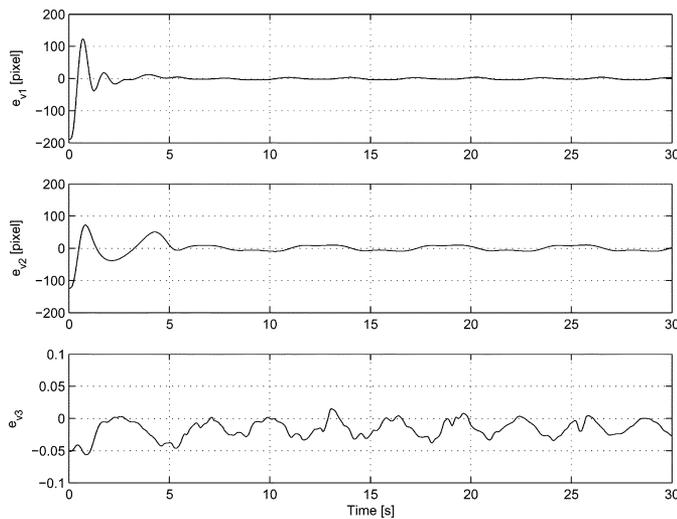


Fig. 11. Error between the actual translation trajectory and the desired translation trajectory given in Fig. 3 for the noise-injected case.

The resulting translational and rotational errors of the target are depicted in Figs. 11 and 12, respectively. The parameter estimate signals are depicted in Figs. 13 and 14. The control input velocities $\omega_e(t)$ and $v_e(t)$ defined in (28) and (29) are depicted in Figs. 15 and 16.

Another simulation was also performed to test the robustness of the controller with respect to the constant rotation matrix R^* . The constant rotation matrix R^* in (5) is coarsely calibrated as $\text{diag}\{1, -1, -1\}$. The resulting translational and rotational errors of the target are depicted in Figs. 17 and 18, respectively.

IX. CONCLUSION

In this brief, an adaptive visual servo controller is developed for the fixed camera configuration to enable the end-effector of a robot manipulator to track a desired trajectory determined by an *a priori* available sequence of images. The controller is formulated using a hybrid composition of image-space pixel information and reconstructed Euclidean information that is obtained via projective homography relationships between the ac-

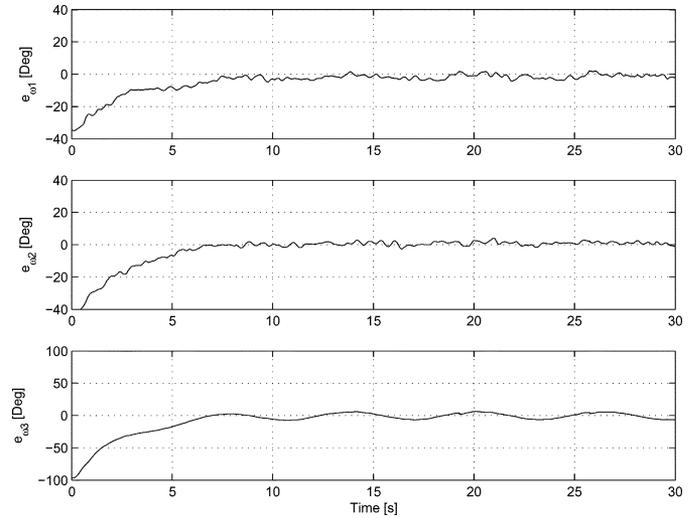


Fig. 12. Error between the actual rotation trajectory and the desired rotation trajectory given in Fig. 4 for the noise-injected case.

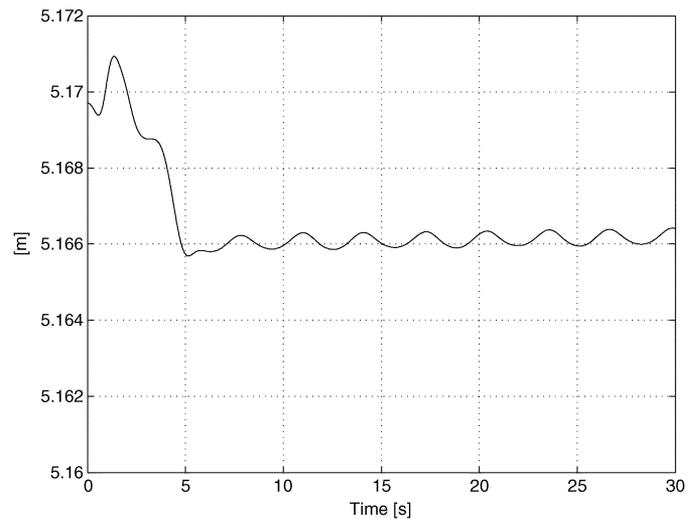


Fig. 13. Parameter estimate for z_1^* for the noise-injected case.

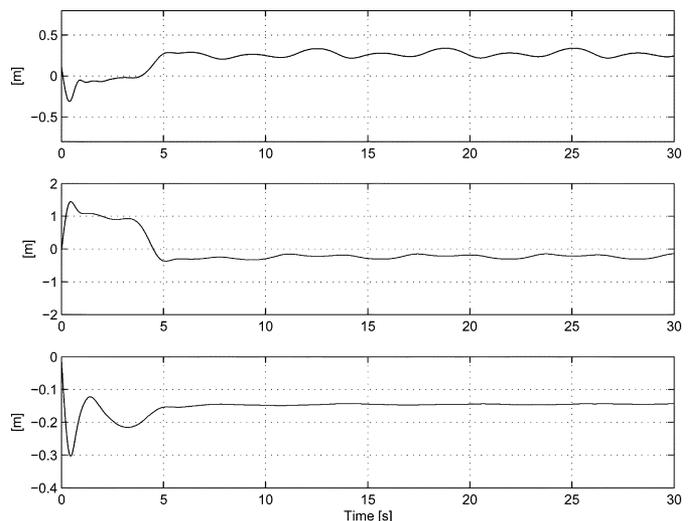


Fig. 14. Parameter estimates for s_1 for the noise-injected case.

tual image, a reference image, and the desired image. To achieve the objective, a Lyapunov-based adaptive control strategy is

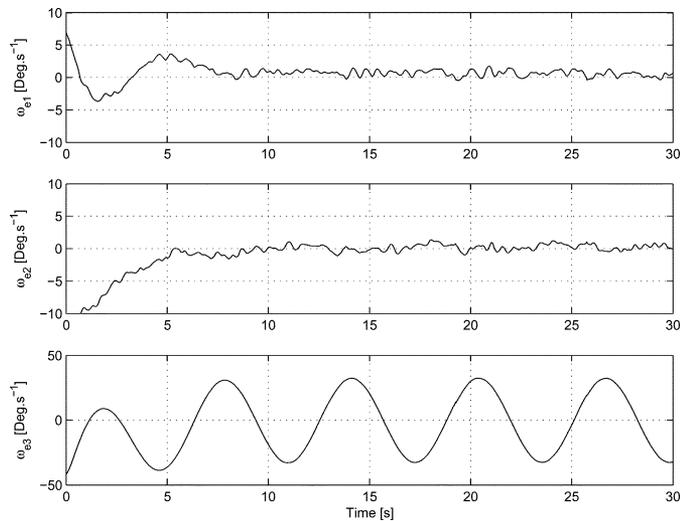


Fig. 15. Angular velocity control input for the noise-injected case.

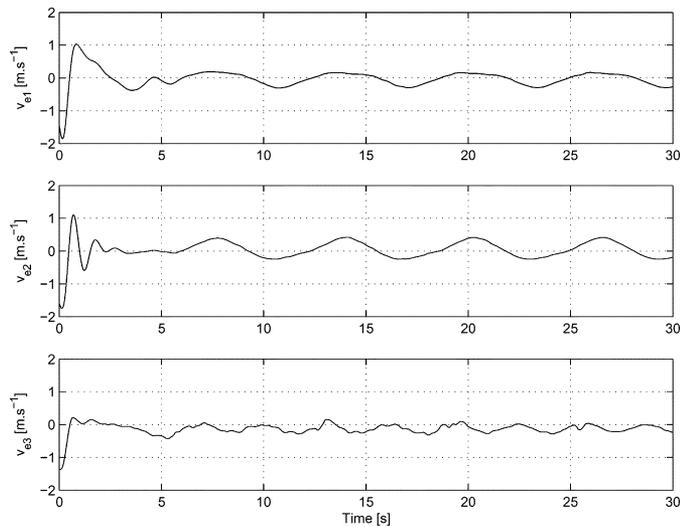
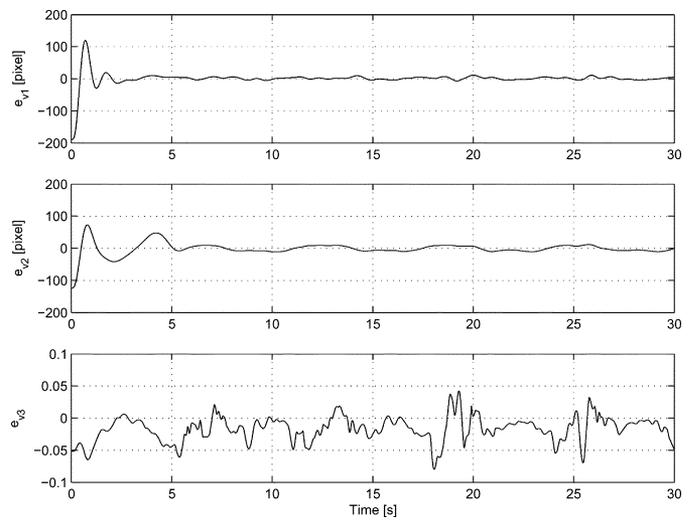
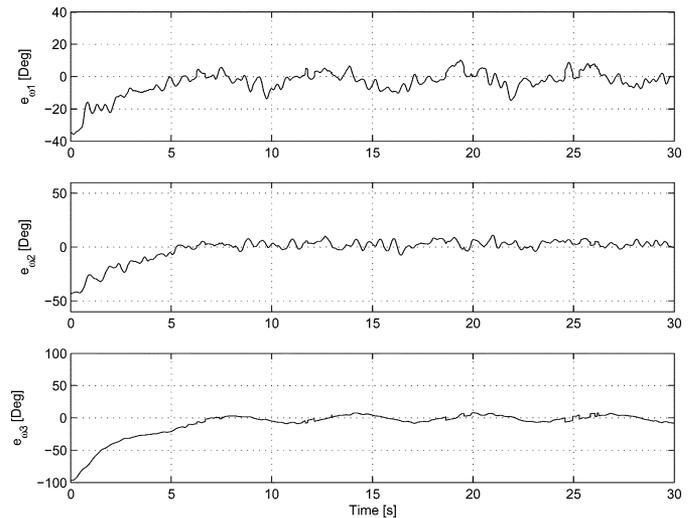


Fig. 16. Linear velocity control input for the noise-injected case.

employed to actively compensate for the lack of unknown depth measurements and unknown object model parameters. Based on the development for the fixed camera controller, an extension is provided to enable a camera held by a robot end-effector to track a desired trajectory determined from a sequence of images (i.e., camera-in-hand tracking). Simulation results were provided to demonstrate the performance of the controller for the fixed camera problem.

APPENDIX A

To obtain geometric insight into the structure of $\bar{R}(t)$ and $\bar{x}_f(t)$ defined in (5) can be obtained from Fig. 1 by placing a fictitious camera that has a frame \mathcal{I}^* attached to its center such that \mathcal{I}^* initially coincides with \mathcal{I} . Since \mathcal{I} and \mathcal{I}^* coincide, the relationship between \mathcal{I}^* and \mathcal{F}^* can be denoted by rotational and translational parameters (x_f^*, R^*) as is evident from Fig. 1. Without relative translational or rotational motion between \mathcal{I}^* and \mathcal{F}^* , the two coordinate frames are moved until \mathcal{F}^* aligns with \mathcal{F} , resulting in Fig. 19. It is now evident that the fixed camera problem reduces to a stereo vision problem with the pa-


 Fig. 17. Error between the actual translation trajectory and the desired translation trajectory given in Fig. 3 for the noise-injected case with a coarse calibration of R^* .

 Fig. 18. Error between the actual rotation trajectory and the desired rotation trajectory given in Fig. 4 for the noise-injected case with a coarse calibration of R^* .

rameters $(x_f - R(R^*)^T x_f^*, R(R^*)^T)$ denoting the translation and rotation between \mathcal{I} and \mathcal{I}^* .

APPENDIX B

Based on the previous definitions for $\omega_e(t)$ and $R(t)$, the following property can be determined [25]:

$$[R\omega_e]_{\times} = \dot{R}R^T. \quad (61)$$

From (5) and (61), the following relationship can be determined:

$$[R\omega_e]_{\times} = \dot{\bar{R}}\bar{R}^T. \quad (62)$$

While several parameterizations can be used to express $\bar{R}(t)$ in terms of $u(t)$ and $\theta(t)$, the open-loop error system for $e_{\omega}(t)$ is derived based on the following exponential parameterization [25]:

$$\bar{R} = \exp([\Theta]_{\times}) = I_3 + \sin \theta [u]_{\times} + 2 \sin^2 \frac{\theta}{2} [u]_{\times}^2 \quad (63)$$

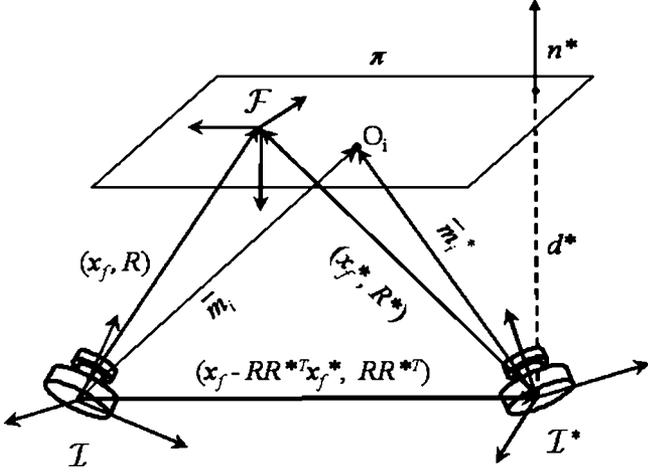


Fig. 19. Geometric relationships for $\bar{R}(t)$ and $\bar{x}_f(t)$.

where the notation I_i denotes an $i \times i$ identity matrix, and the notation $[u]_{\times}$ denotes the skew-symmetric matrix form of $u(t)$. The parameterization $\Theta_d(t)$ can be related to $\bar{R}_d(t)$ as follows:

$$\bar{R}_d = \exp([\Theta_d]_{\times}) = I_3 + \sin \theta_d [u_d]_{\times} + 2 \sin^2 \frac{\theta_d}{2} [u_d]_{\times}^2. \quad (64)$$

To facilitate the development of the open-loop dynamics for $e_{\omega}(t)$, the expression developed in (62) can be used along with (63) and the time derivative of (63), to obtain the following expression:

$$[R\omega_e]_{\times} = \sin \theta [\dot{u}]_{\times} + [u]_{\times} \dot{\theta} + (1 - \cos \theta) [[u]_{\times} \dot{u}]_{\times} \quad (65)$$

where the following properties were utilized [15], [16]:

$$[u]_{\times} \zeta = -[\zeta]_{\times} u \quad (66)$$

$$[u]_{\times}^2 = uu^T - I_3 \quad (67)$$

$$[u]_{\times} uu^T = 0 \quad (68)$$

$$[u]_{\times} [\dot{u}]_{\times} [u]_{\times} = 0 \quad (69)$$

$$[[u]_{\times} \dot{u}]_{\times} = [u]_{\times} [\dot{u}]_{\times} - [\dot{u}]_{\times} [u]_{\times}. \quad (70)$$

To facilitate further development, the time derivative of (19) is determined as follows:

$$\dot{\Theta} = \dot{u}\theta + u\dot{\theta}. \quad (71)$$

After multiplying (71) by $(I_3 + [u]_{\times}^2)$, the following expression can be obtained:

$$(I_3 + [u]_{\times}^2) \dot{\Theta} = u\dot{\theta} \quad (72)$$

where (67) and the following properties were utilized:

$$u^T u = 1 \quad u^T \dot{u} = 0. \quad (73)$$

Likewise, by multiplying (71) by $-[u]_{\times}^2$ and then utilizing (73) the following expression is obtained:

$$-[u]_{\times}^2 \dot{\Theta} = \dot{u}\theta. \quad (74)$$

From the expression in (65), the properties given in (66), (71), (72), (74), and the fact that

$$\sin^2 \theta = \frac{1}{2}(1 - \cos 2\theta)$$

can be used to obtain the following expression:

$$R\omega_e = L_{\omega}^{-1} \dot{\Theta} \quad (75)$$

where $L_{\omega}(t)$ is defined in (23). After multiplying both sides of (75) by $L_{\omega}(t)$, the open-loop dynamics for $\dot{\Theta}$ can be obtained. After substituting (75) into the time derivative of (18), the open-loop dynamics for $e_{\omega}(t)$ given by (22) can be obtained.

To develop the open-loop error system for $e_v(t)$, the time derivative of (16) is obtained as follows:

$$\dot{e}_v = \dot{p}_e - \dot{p}_{ed} = \frac{1}{z_1} A_e L_v \dot{m}_1 - \dot{p}_{ed} \quad (76)$$

where (1), (7), (14), (17), and the definition of $\alpha_i(t)$ in (10) were utilized. After taking the time derivative of the first equation in (3), $\dot{m}_1(t)$ can be determined as follows:

$$\dot{m}_1 = Rv_e + R[\omega_e]_{\times} s_1 \quad (77)$$

where (61) and the following property have been utilized [15]:

$$[R\omega_e]_{\times} = R[\omega_e]_{\times} R^T. \quad (78)$$

After substituting (77) into (76), multiplying the resulting expression by z_1^* , and utilizing the definition of $\alpha_i(t)$ in (10), the open-loop error system given in (25) is obtained.

APPENDIX C

As stated in Section IV, to ensure that $\bar{m}_i(t)$ tracks $\bar{m}_{di}(t)$ from the Euclidean reconstruction given in (1), the tracking control objective can be stated as follows: $\bar{R}(t) \rightarrow \bar{R}_d(t)$, $m_1(t) \rightarrow m_{d1}(t)$, and $z_1(t) \rightarrow z_{d1}(t)$. This appendix describes how this objective can be achieved provided the result given in (35) is obtained. To this end, the expressions in (13) and (17) can be used to conclude that if $\|e_v(t)\| \rightarrow 0$ then $p_1(t) \rightarrow p_{d1}(t)$ and the ratio $\alpha_1(t)/\alpha_{d1}(t) \rightarrow 1$. Hence, from (14) and the definition of the depth ratios in (10) and (11), it can be shown that $m_1(t) \rightarrow m_{d1}(t)$ and $z_1(t) \rightarrow z_{d1}(t)$. Given that $m_1(t) \rightarrow m_{d1}(t)$ and $z_1(t) \rightarrow z_{d1}(t)$, then (7) and (8) can be used to prove that $\bar{m}_1(t) \rightarrow \bar{m}_{d1}(t)$.

To examine if $\bar{R}(t) \rightarrow \bar{R}_d(t)$, we first take the difference between the expressions defined in (63) and (64) can be determined as follows:

$$\bar{R} - \bar{R}_d = \sin \theta [u]_{\times} - \sin \theta_d [u_d]_{\times} + 2 \sin^2 \frac{\theta}{2} [u]_{\times}^2 - 2 \sin^2 \frac{\theta_d}{2} [u_d]_{\times}^2. \quad (79)$$

To continue the analysis, we can see from the result in (35) that $\|e_{\omega}(t)\| \rightarrow 0$ and, hence, we can use (18) and (19) to show that

$$u(t)\theta(t) \rightarrow u_d(t)\theta_d(t) \text{ as } t \rightarrow \infty. \quad (80)$$

To see if (80) implies that $\bar{R}(t) \rightarrow \bar{R}_d(t)$, we note that (35) and (80) imply that

$$\|u(t)\theta(t)\|^2 \rightarrow \|u_d(t)\theta_d(t)\|^2 \text{ as } t \rightarrow \infty \quad (81)$$

which implies that

$$\theta^2(t) \|u(t)\|^2 \rightarrow \theta_d^2(t) \|u_d(t)\|^2 \text{ as } t \rightarrow \infty. \quad (82)$$

Since $\|u(t)\| = \|u_d(t)\| = 1$, we can see from (82) that

$$\theta(t) \rightarrow \pm\theta_d(t) \text{ as } t \rightarrow \infty. \quad (83)$$

We can now see from (80) that

$$\begin{aligned} \text{Case 1) } & u(t) \rightarrow u_d(t) \text{ when } \theta(t) \rightarrow \theta_d(t) \\ \text{Case 2) } & u(t) \rightarrow -u_d(t) \text{ when } \theta(t) \rightarrow -\theta_d(t). \end{aligned} \quad (84)$$

After substituting each case given in (84) into (79) and then passing the limit, it is clear that $\bar{R}(t) \rightarrow \bar{R}_d(t)$. Based on the results that $\bar{m}_{11}(t) \rightarrow \bar{m}_{d1}(t)$ and that $\bar{R}(t) \rightarrow \bar{R}_d(t)$, it is clear that $\bar{m}_{vi}(t) \rightarrow \bar{m}_{di}(t)$.

REFERENCES

- [1] F. Chaumette and E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2000, pp. 630–635.
- [2] F. Chaumette, E. Malis, and S. Boudet, "2D 1/2 visual servoing with respect to a planar object," in *Proc. Workshop New Trends in Image-Based Robot Servicing*, 1997, pp. 45–52.
- [3] F. Conticelli and B. Allotta, "Nonlinear controllability and stability analysis of adaptive image-based systems," *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, Apr. 2001.
- [4] F. Conticelli and B. Allotta, "Discrete-time robot visual feedback in 3-D positioning tasks with depth adaptation," *IEEE/ASME Trans. Mechatronics*, vol. 6, no. 3, pp. 356–363, Sep. 2001.
- [5] P. I. Corke and S. A. Hutchinson, "A new hybrid image-based visual servo control scheme," in *Proc. IEEE Conf. Decision and Control*, 2000, pp. 2521–2527.
- [6] N. J. Cowan and D. Koditschek, "Planar image-based visual servoing as a navigation problem," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, Apr. 2000, pp. 1720–1725.
- [7] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. Robot. Autom.*, vol. 18, pp. 521–533, Aug. 2002.
- [8] K. Deguchi, "Optimal motion control for image-based visual servoing by decoupling translation and rotation," in *Proc. Int. Conf. Intelligent Robots and Systems*, Oct. 1998, pp. 705–711.
- [9] M. de Queiroz, D. Dawson, S. Nagarkatti, and F. Zhang, *Lyapunov-Based Control of Mechanical Systems*. Cambridge, MA: Birkhauser, 2000.
- [10] Y. Fang, A. Behal, W. E. Dixon, and D. M. Dawson, "Adaptive 2.5D visual servoing of kinematically redundant robot manipulators," in *Proc. Conf. Decision Control*, Las Vegas, NV, Dec. 2002, pp. 2860–2865.
- [11] Y. Fang, D. M. Dawson, W. E. Dixon, and M. S. de Queiroz, "2.5D visual servoing of wheeled mobile robots," in *Proc. Conf. Decision Control*, Las Vegas, NV, Dec. 2002, pp. 2866–2871.
- [12] Y. Fang, W. E. Dixon, D. M. Dawson, and J. Chen, "Robust 2.5D visual servoing for robot manipulators," in *Proc. IEEE Amer. Control Conf.*, Denver, CO, Jun. 2003, pp. 3311–3316.
- [13] O. Faugeras, *Three-Dimensional Computer Vision*. Cambridge, MA: MIT Press, 2001.
- [14] O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 2, no. 3, pp. 485–508, 1988.
- [15] C. A. Felippa, *A Systematic Approach to the Element-Independent Corotational Dynamics of Finite Elements*, College Eng., Univ. Colorado, Boulder, CO, Jan. 2000.
- [16] E. Malis, "Contributions à la modélisation et à la commande en asservissement visuel," Ph.D. dissertation, Univ. Rennes I, IRISA, Rennes, France, Nov. 1998.
- [17] E. Malis and F. Chaumette, "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 176–186, Apr. 2002.
- [18] ———, "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *Int. J. Comput. Vis.*, vol. 37, no. 1, pp. 79–97, Jun. 2000.
- [19] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.
- [20] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 534–549, Aug. 2002.
- [21] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, pp. 501–518, Oct. 1992.
- [22] A. Ruf and R. Horaud, "Visual trajectories from uncalibrated stereo," in *Proc. IEEE Int. Conf. Intelligent Robots Systems*, Sep. 1997, pp. 83–91.
- [23] R. Singh, R. M. Voyle, D. Littau, and N. P. Papanikolopoulos, "Alignment of an eye-in-hand system to real objects using virtual images," in *Proc. Workshop Robust Vision, Vision-based Control of Motion*, *IEEE Int. Conf. Robotics Automation*, May 1998.
- [24] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [25] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: Wiley, 1989.
- [26] Z. Zhang and A. R. Hanson, "Scaled Euclidean 3-D reconstruction based on externally uncalibrated cameras," in *Proc. IEEE Symp. Computer Vision*, 1995, pp. 37–42.