Available online at www.sciencedirect.com
ScienceDirect
Automatica 43 (2007) 1165–1177
automatica
www.elsevier.com/locate/automatica

Navigation function-based visual servo control☆

Jian Chen[a,*], Darren M. Dawson[b], Warren E. Dixon[c], Vilas K. Chitrakaran[b]

[a]Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109, USA
[b]Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29631, USA
[c]Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA

Received 29 October 2004; received in revised form 23 October 2006; accepted 28 December 2006
Available online 18 May 2007

Abstract

In this paper, the mapping between the desired camera feature vector and the desired camera pose (i.e., the position and orientation) is investigated to develop a measurable image Jacobian-like matrix. An image-space path planner is then proposed to generate a desired image trajectory based on this measurable image Jacobian-like matrix and an image-space navigation function (NF) (i.e., a special potential field function) while satisfying rigid body constraints. An adaptive, homography-based visual servo tracking controller is then developed to navigate the position and orientation of a camera held by the end-effector of a robot manipulator to a goal position and orientation along the desired image-space trajectory while ensuring the target points remain visible (i.e., the target points avoid self-occlusion and remain in the field-of-view (FOV)) under certain technical restrictions. Due to the inherent nonlinear nature of the problem and the lack of depth information from a monocular system, a Lyapunov-based analysis is used to analyze the path planner and the adaptive controller. Simulation results are provided to illustrate the performance of the proposed approach.

Keywords: Visual servo control; Robot navigation; Lyapunov technique

1. Introduction

There is significant motivation to provide improved autonomy for robotic systems. In part, this motivation has led researchers to investigate the basic science challenges leading to the development of visual servo controllers as a means to provide improved robot autonomy. In general, visual servo controllers can be divided into position-based visual servo (PBVS) control, image-based visual servo (IBVS), and hybrid approaches. PBVS is based on the idea of using a vision system to reconstruct the Euclidean space and then developing the servo controller on the reconstructed information. A well-known issue with this strategy is that the target object may exit the camera field-of-view (FOV). IBVS control is based on the idea of directly servoing on the image-space information, with reported advantages of increased robustness to camera calibration and improved capabilities to ensure the target remains visible. Even for IBVS controllers that are formulated as regulation controllers, if the initial error is large then excessive control action and transient response can cause the target to leave the FOV, and may lead to trajectories that are not physically valid or optimal due to the nonlinearities and potential singularities associated with the transformation between the image space and the Euclidean space (Chaumette, 1998). For a review of IBVS and PBVS controllers see Hager and Hutchinson (1996).

In light of the characteristics of IBVS and PBVS, several researchers have recently explored hybrid approaches. For example, homography-based visual servo control techniques (coined 2.5D controllers) have been recently developed in a series of papers by Malis and Chaumette (e.g., Chaumette, Malis, & Boudet, 1997; Malis & Chaumette, 2000; Malis, Chaumette, & Boudet, 1999). The homography-based approach exploits a combination of reconstructed Euclidean

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Thomas Parisini under the direction of Editor Miroslav Krstic.

* Corresponding author. Tel.: +1 734 355 2419; fax: +1 302 370 6417.
E-mail address: jianc@umich.edu (J. Chen).

0005-1098/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2006.12.018

information and image-space information in the control design. The Euclidean information is reconstructed by decoupling the interaction between translational and rotational components of a homography matrix. As stated in Malis et al. (1999), some advantages of this methodology over the aforementioned IBVS and PBVS approaches are that an accurate Euclidean model of the environment (or target object) is not required, and potential singularities in the image Jacobian are eliminated (i.e., the image Jacobian for homography-based visual servo controllers is typically triangular). Motivated by the advantages of the homography-based strategy, several researchers have recently developed various regulation controllers for robot manipulators (see Chen, Dawson, Dixon, & Behal, 2005; Corke & Hutchinson, 2000; Deguchi, 1998).

While homography-based approaches exploit the advantages of IBVS and PBVS, a common problem with all the aforementioned approaches is the inability to achieve the control objective while ensuring the target features remain visible. To address this issue, Mezouar and Chaumette developed a path-following IBVS algorithm in Mezouar and Chaumette (2002) where the path to a goal point is generated via a potential function that incorporates motion constraints; however, as stated in Mezouar and Chaumette (2002), local minima associated with traditional potential functions may exist. Using a specialized potential function (coined as navigation function (NF)) originally proposed in Koditschek and Rimon (1990) and Rimon and Koditschek (1992), Cowan et al. developed a hybrid position/image-space controller that forces a manipulator to a desired setpoint while ensuring the object remains visible (i.e., the NF ensures no local minima) and by avoiding pitfalls such as self-occlusion (Cowan, Weingarten, & Koditscheck, 2002). However, as stated in Mezouar and Chaumette (2002), this approach requires the complete knowledge of the space topology and requires an object model. In Gans and Hutchinson (2003), Gans and Hutchinson developed a strategy that switches between an IBVS and a PBVS controller to ensure asymptotic stability of the position and orientation (i.e., pose) in the Euclidean and image space. An image-space based follow-the-leader application for mobile robots was developed in Cowan, Shakernia, Vidal, and Sastry (2003) that exploits an image-space NF. Specifically, an input/output feedback linearization technique is applied to the mobile robot kinematic model to yield a controller that yields "string stability" (Fierro, Song, Das, & Kumar, 2002, Chapter 5). Without a feedforward component, the controller in Cowan et al. (2003) yields an approximate "input-to-formation" stability (i.e., a local, linear exponential system with a bounded disturbance). An NF-based approach to the follow-the-leader problem for a group of fully actuated holonomic mobile robots is considered in Pereira, Das, Kumar, and Campos (2003) where configuration-based constraints are developed to ensure the robot edges remain in the sight of an omnidirectional camera. While a Lyapunov-based analysis is provided in Pereira et al. (2003) to ensure that the NF decreases to the goal position, the stability of the overall system is not examined.

Motivated by the image-space NF developed in Cowan et al. (2002), an off-line desired image trajectory generator is proposed based on a new image Jacobian-like matrix for the monocular, camera-in-hand problem. This approach generates a desired camera pose trajectory that moves the camera from the initial camera pose to a goal camera pose while ensuring that all the feature points of the object remain visible under certain technical restrictions. To develop a desired camera pose trajectory that ensures all feature points remain visible, a unique relationship is formulated between the desired image feature vector and the desired camera pose. The resulting image Jacobian-like matrix is related to the camera pose, rather than the camera velocity as in other approaches (Chaumette, 1998). Motivation for the development of this relationship is that the resulting image Jacobian-like matrix is measurable, and, hence, does not suffer from the lack of robustness associated with estimation-based methods. Furthermore, the desired image generated with this image Jacobian-like matrix satisfies rigid body constraints (the terminology, rigid body constraints, in this paper is utilized to denote the image feature vector constraints in which feature points have a fixed relative position to each other in Euclidean space). Building on our recent research in Chen et al. (2005), an adaptive homography-based visual tracking controller is then developed to ensure that the actual camera pose tracks the desired camera pose trajectory (i.e., the actual features track the desired feature point trajectory) despite the fact that time-varying depth from the camera to the reference image plane is not measurable from the monocular camera system. Based on the analysis of the homography-based controller, bounds are developed that can be used to ensure that the actual image features also remain visible under certain technical restrictions. A Lyapunov-based analysis is provided to support the claims for the path planner and to analyze the stability of the adaptive tracking controller. Simulation results are provided to illustrate the performance of the proposed approach.

2. Geometric modeling

2.1. Euclidean homography

Four feature points, denoted by $O_i \ \forall i = 1, 2, 3, 4$ are assumed to be located on a reference plane π (see Fig. 1), and are considered to be coplanar[1] and not colinear. The reference plane can be related to the coordinate frames $\mathcal{F}$, $\mathcal{F}_d$, and $\mathcal{F}^*$ depicted in Fig. 1 that denote the actual, desired, and goal pose of the camera, respectively.

Specifically, the following relationships can be developed from the geometry between the coordinate frames and the feature points located on π:

$$\bar{m}_i = x_f + R\bar{m}_i^*, \quad \bar{m}_{di} = x_{fd} + R_d\bar{m}_i^*, \tag{1}$$

where $\bar{m}_i(t) \triangleq [x_i(t) \ y_i(t) \ z_i(t)]^{\mathrm{T}}$, $\bar{m}_{di}(t) \triangleq [x_{di}(t) \ y_{di}(t) \ z_{di}(t)]^{\mathrm{T}}$, and $\bar{m}_i^* \triangleq [x_i^* \ y_i^* \ z_i^*]^{\mathrm{T}}$ denote the Euclidean coordinates

[1] It should be noted that if four coplanar target points are not available then the subsequent development can exploit the classic eight-points algorithm (Malis & Chaumette, 2000) with no four of the eight target points being coplanar.

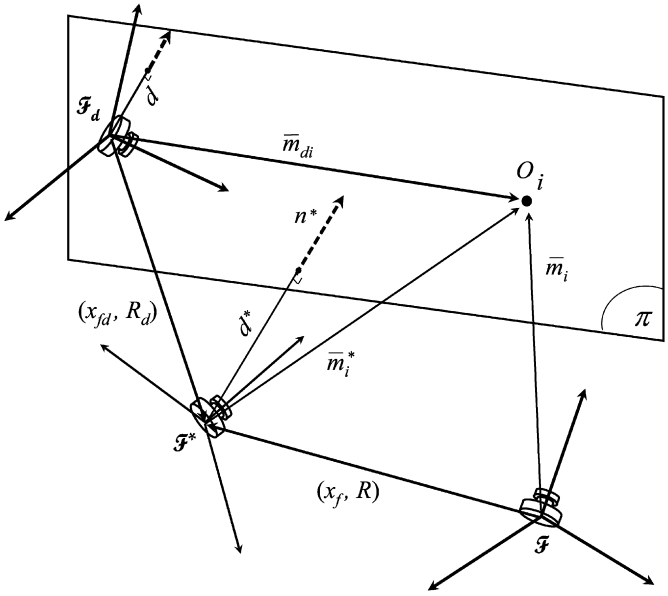

Fig. 1. Coordinate frame relationships.

of O_i expressed in $\mathcal{F}$, $\mathcal{F}_d$, and $\mathcal{F}^*$, respectively. In (1), $R(t)$, $R_d(t) \in SO(3)$ 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. The constant unknown distance from the origin of $\mathcal{F}^*$ to π is denoted by $d^* \in \mathbb{R}$ and is defined as follows:

$$d^* = n^{*\mathrm{T}} \bar{m}_i^*, \tag{2}$$

where $n^* \in \mathbb{R}^3$ denotes the constant unit normal to the plane π expressed in the coordinates of $\mathcal{F}^*$. Also from Fig. 1, the unknown, time-varying distance from the origin of $\mathcal{F}_d$ to π, denoted by $d(t) \in \mathbb{R}$, can be expressed as follows:

$$d = n^{*\mathrm{T}} R_d^{\mathrm{T}} \bar{m}_{di}. \tag{3}$$

Since the Euclidean position of $\mathcal{F}$, $\mathcal{F}_d$, and $\mathcal{F}^*$ cannot be directly measured, the expressions in (1) need to be related to the measurable image-space coordinates. To this end, the normalized Euclidean coordinates of O_i expressed in terms of $\mathcal{F}$, $\mathcal{F}_d$, and $\mathcal{F}^*$ as $m_i(t)$, $m_{di}(t)$, $m_i^* \in \mathbb{R}^3$, respectively, are defined as follows:

$$m_i \triangleq \frac{\bar{m}_i}{z_i}, \quad m_{di} \triangleq \frac{\bar{m}_{di}}{z_{di}}, \quad m_i^* \triangleq \frac{\bar{m}_i^*}{z_i^*} \tag{4}$$

under the standard assumption that $z_i(t)$, $z_{di}(t)$, $z_i^* > \varepsilon$ where ε denotes an arbitrarily small positive constant. Based on (2) and (4), the expression in (1) can be rewritten as follows:

$$m_i = \underbrace{\frac{z_i^*}{z_i}}_{\alpha_i} \underbrace{\left(R + \frac{x_f}{d^*} n^{*\mathrm{T}}\right)}_{H} m_i^*, \tag{5}$$

$$m_{di} = \underbrace{\frac{z_i^*}{z_{di}}}_{\alpha_{di}} \underbrace{\left(R_d + \frac{x_{fd}}{d^*} n^{*\mathrm{T}}\right)}_{H_d} m_i^*, \tag{6}$$

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 (Faugeras, 2001).

2.2. Projective homography

Each feature point on π has projected pixel coordinates denoted by $u_i(t)$, $v_i(t) \in \mathbb{R}$ in $\mathcal{F}$, $u_{di}(t)$, $v_{di}(t) \in \mathbb{R}$ in $\mathcal{F}_d$, and u_i^*, $v_i^* \in \mathbb{R}$ in $\mathcal{F}^*$ that are defined as follows:

$$p_i \triangleq [u_i \;\; v_i \;\; 1]^{\mathrm{T}}, \quad p_{di} \triangleq [u_{di} \;\; v_{di} \;\; 1]^{\mathrm{T}},$$
$$p_i^* \triangleq [u_i^* \;\; v_i^* \;\; 1]^{\mathrm{T}}. \tag{7}$$

In (7), $p_i(t)$, $p_{di}(t)$, $p_i^* \in \mathbb{R}^3$ represent the image-space coordinates of the time-varying feature points, the desired time-varying feature point trajectory, and the constant reference feature points, respectively. To calculate the Euclidean homography given in (5) and (6) 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 (Faugeras, 2001):

$$p_i = A m_i, \quad p_{di} = A m_{di}, \quad p_i^* = A m_i^*, \tag{8}$$

where $A \in \mathbb{R}^{3\times 3}$ is a known, constant, and invertible intrinsic camera calibration matrix with the following form:

$$A = \begin{bmatrix} a_1 & a_2 & a_4 \\ 0 & a_3 & a_5 \\ 0 & 0 & 1 \end{bmatrix}, \tag{9}$$

where $a_i \in \mathbb{R}$ $\forall i = 1, 2, \ldots, 5$ denote known, constant calibration parameters. After substituting (8) into (5) and (6), the following relationships can be developed:

$$p_i = \alpha_i \underbrace{(AHA^{-1})}_{G} p_i^*, \quad p_{di} = \alpha_{di} \underbrace{(AH_dA^{-1})}_{G_d} p_i^*, \tag{10}$$

where $G(t)$, $G_d(t) \in \mathbb{R}^{3\times 3}$ denote projective homographies. Given the images of the four feature points on π expressed in $\mathcal{F}$, $\mathcal{F}_d$, and $\mathcal{F}^*$, a linear system of equations can be developed from (10). From the linear system of equations, a decomposition algorithm (e.g., the Faugeras decomposition algorithm in Faugeras (2001)) can be used to compute $\alpha_i(t)$, $\alpha_{di}(t)$, n^*, $R(t)$, and $R_d(t)$ (see Chen et al., 2005 for details).[2] Hence, $\alpha_i(t)$, $\alpha_{di}(t)$, n^*, $R(t)$, and $R_d(t)$ are known signals that can be used in the subsequent development.

2.3. Kinematic model of vision system

The camera pose, denoted by $\Upsilon(t) \in \mathbb{R}^6$, can be expressed in terms of a hybrid of pixel and reconstructed Euclidean information as follows:

$$\Upsilon(t) \triangleq [p_{e1}^{\mathrm{T}} \;\; \Theta^{\mathrm{T}}]^{\mathrm{T}}, \tag{11}$$

[2] The initial best guess of n^* can be utilized to resolve the decomposition ambiguity. See Chen, Dixon, Dawson, and McIntire (2006) for details.

where the extended pixel coordinate $p_{e1}(t) \in \mathbb{R}^3$ is defined as follows:

$$p_{e1} = [u_1 \;\; v_1 \;\; -\ln(\alpha_1)]^T, \tag{12}$$

and $\Theta(t) \in \mathbb{R}^3$ denotes the following axis-angle representation of $R(t)$ (see Chen et al., 2005 for details):

$$\Theta = \mu(t)\theta(t). \tag{13}$$

In (12), $\ln(\cdot)$ denotes the natural logarithm, and $\alpha_1(t)$ is introduced in (5). In (13), $\mu(t) \in \mathbb{R}^3$ represents the unit axis of rotation, and $\theta(t)$ denotes the rotation angle about that axis. Based on the development in Appendix A, the open-loop dynamics for $\Upsilon(t)$ can be expressed as follows:

$$\dot{\Upsilon} = \begin{bmatrix} \dot{p}_{e1} \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{z_1} A_{e1} & A_{e1}[m_1]_\times \\ 0 & -L_\omega \end{bmatrix} \begin{bmatrix} v_c \\ \omega_c \end{bmatrix}, \tag{14}$$

where $v_c(t) \in \mathbb{R}^3$ and $\omega_c(t) \in \mathbb{R}^3$ denote the linear and angular velocity of the camera expressed in terms of $\mathcal{F}$, $A_{ei}(u_i, v_i) \in \mathbb{R}^{3\times 3}$ is a known, invertible matrix defined as follows:

$$A_{ei} = A - \begin{bmatrix} 0 & 0 & u_i \\ 0 & 0 & v_i \\ 0 & 0 & 0 \end{bmatrix}, \quad i = 1, 2, 3, 4, \tag{15}$$

and the invertible Jacobian-like matrix $L_\omega(\theta, \mu) \in \mathbb{R}^{3\times 3}$ is defined as (Malis et al., 1999)

$$L_\omega = I_3 - \frac{\theta}{2}[\mu]_\times + \left(1 - \frac{\operatorname{sinc}(\theta)}{\operatorname{sinc}^2(\theta/2)}\right)[\mu]_\times^2, \tag{16}$$

where $I_n \in \mathbb{R}^{n\times n}$ denotes the $n \times n$ identity matrix, $[\mu_p]_\times$ denotes the 3×3 skew-symmetric expansion of $\mu_p(t)$, and

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

Remark 1. As stated in Spong and Vidyasagar (1989), the axis-angle representation of (13) is not unique, in the sense that a rotation of $-\theta(t)$ about $-\mu(t)$ is equal to a rotation of $\theta(t)$ about $\mu(t)$. A particular solution for $\theta(t)$ and $\mu(t)$ can be determined as follows (Spong & Vidyasagar, 1989):

$$\theta_p = \cos^{-1}\left(\frac{1}{2}(\operatorname{tr}(R) - 1)\right), \quad [\mu_p]_\times = \frac{R - R^T}{2\sin(\theta_p)}, \tag{17}$$

where the notation $\operatorname{tr}(\cdot)$ denotes the trace of a matrix. From (17), it is clear that

$$0 \leqslant \theta_p(t) \leqslant \pi. \tag{18}$$

3. Image-based path planning

The path planning objective involves regulating the pose of a camera held by the end-effector of a robot manipulator to a desired camera pose along an image-space trajectory while ensuring the target points remain visible. To achieve this objective, a desired camera pose trajectory is constructed in this section so that the desired image feature vector, denoted by $\bar{p}_d(t) \triangleq [u_{d1}(t) \; v_{d1}(t) \; \ldots \; u_{d4}(t) \; v_{d4}(t)]^T \in \mathbb{R}^8$, remains in a set, denoted by $\mathcal{D} \subset \mathbb{R}^8$, where all four feature points of the target remain visible for a valid camera pose. The constant, goal image feature vector $\bar{p}^* \triangleq [u_1^* \; v_1^* \; \ldots \; u_4^* \; v_4^*]^T \in \mathbb{R}^8$ is assumed be in the interior of $\mathcal{D}$. To generate the desired camera pose trajectory such that $\bar{p}_d(t) \in \mathcal{D}$, the special artificial potential function coined as NF in Koditschek and Rimon (1990), can be used. Specifically, the NFs used in this paper are defined as follows (Rimon & Koditschek, 1992).

Definition 1. A map $\varphi(\bar{p}_d) : \mathcal{D} \to [0, 1]$, is an NF if:

(P1) analytic on $\mathcal{D}$ (at least the first and second partial derivatives exist and are bounded on $\mathcal{D}$);
(P2) a unique minimum exists at $\bar{p}^*$;
(P3) it obtains a maximum value on the boundary of $\mathcal{D}$ (i.e., admissible on $\mathcal{D}$);
(P4) it is a Morse function (i.e., the matrix of second partial derivatives, the Hessian, evaluated at its critical points is non-singular (and has bounded elements based on the smoothness property in (P1))).

3.1. Pose space to image space relationship

To develop a desired camera pose trajectory that ensures $\bar{p}_d(t) \in \mathcal{D}$, the desired image feature vector is related to the desired camera pose, denoted by $\Upsilon_d(t) \in \mathbb{R}^6$, through the following relationship:

$$\bar{p}_d = \Pi(\Upsilon_d), \tag{19}$$

where $\Pi(\cdot) : \mathbb{R}^6 \to \mathcal{D}$ denotes an unknown function that maps the camera pose to the image feature vector.[3] In (19), the desired camera pose is defined as follows:

$$\Upsilon_d(t) \triangleq [\, p_{ed1}^T \quad \Theta_d^T \,]^T, \tag{20}$$

where $p_{ed1}(t) \in \mathbb{R}^3$ denotes the desired extended pixel coordinates defined as follows:

$$p_{ed1} = [\, u_{d1} \quad v_{d1} \quad -\ln(\alpha_{d1}) \,]^T, \tag{21}$$

where $\alpha_{d1}(t)$ is introduced in (6), and $\Theta_d(t) \in \mathbb{R}^3$ denotes the axis-angle representation of $R_d(t)$ as follows:

$$\Theta_d = \mu_d(t)\theta_d(t), \tag{22}$$

where $\mu_d(t) \in \mathbb{R}^3$ and $\theta_d(t) \in \mathbb{R}$ are defined in the same manner as $\mu(t)$ and $\theta(t)$ in (13) with respect to $R_d(t)$.

[3] The reason we choose four feature points to construct the image feature vector is that the same image of three points can be seen from four different camera poses (Horaud, 1987). A unique camera pose can theoretically be obtained by using at least four points (Chaumette, 1998). Therefore, the map $\Pi(\cdot)$ is a unique mapping with the image feature vector corresponding to a valid camera pose.

3.2. Desired image trajectory planning

After taking the time derivative of (19), the following expression can be obtained:

$$\dot{\bar{p}}_d = L_{\Upsilon_d}\dot{\Upsilon}_d, \tag{23}$$

where $L_{\Upsilon_d}(\bar{p}_d) \triangleq \partial \bar{p}_d/\partial \Upsilon_d \in \mathbb{R}^{8\times 6}$ denotes an image Jacobian-like matrix. Based on the development in Appendix B, a measurable expression for $L_{\Upsilon_d}(t)$ can be developed as follows:

$$L_{\Upsilon_d} = \bar{I}T, \tag{24}$$

where $\bar{I} \in \mathbb{R}^{8\times 12}$ denotes a constant, row-delete matrix defined as follows:

$$\bar{I} = \begin{bmatrix} I_2 & 0^2 & 0_2 & 0^2 & 0_2 & 0^2 & 0_2 & 0^2 \\ 0_2 & 0^2 & I_2 & 0^2 & 0_2 & 0^2 & 0_2 & 0^2 \\ 0_2 & 0^2 & 0_2 & 0^2 & I_2 & 0^2 & 0_2 & 0^2 \\ 0_2 & 0^2 & 0_2 & 0^2 & 0_2 & 0^2 & I_2 & 0^2 \end{bmatrix},$$

where $0_n \in \mathbb{R}^{n\times n}$ denotes an $n \times n$ matrix of zeros, $0^n \in \mathbb{R}^n$ denotes an $n \times 1$ column of zeros, and $T(t) \in \mathbb{R}^{12\times 6}$ is a measurable auxiliary matrix defined as follows:

$$T = \begin{bmatrix} I_3 & 0_3 \\ \dfrac{\beta_1}{\beta_2}A_{ed2}A_{ed1}^{-1} & A_{ed2}\left[\dfrac{\beta_1}{\beta_2}m_{d1} - m_{d2}\right]_\times L_{\omega d}^{-1} \\ \dfrac{\beta_1}{\beta_3}A_{ed3}A_{ed1}^{-1} & A_{ed3}\left[\dfrac{\beta_1}{\beta_3}m_{d1} - m_{d3}\right]_\times L_{\omega d}^{-1} \\ \dfrac{\beta_1}{\beta_4}A_{ed4}A_{ed1}^{-1} & A_{ed4}\left[\dfrac{\beta_1}{\beta_4}m_{d1} - m_{d4}\right]_\times L_{\omega d}^{-1} \end{bmatrix}. \tag{25}$$

In (25), $A_{edi}(u_{di}, v_{di}) \in \mathbb{R}^{3\times 3}$ and the Jacobian-like matrix $L_{\omega d}(\theta_d, \mu_d) \in \mathbb{R}^{3\times 3}$ are defined as in (15) and (16) with respect to $u_{di}(t)$, $v_{di}(t)$, $\mu_d(t)$, and $\theta_d(t)$, respectively. The auxiliary variable $\beta_i(t) \in \mathbb{R}$ in (25) is defined as follows:

$$\beta_i \triangleq \frac{z_{di}}{d}, \quad i = 1, 2, 3, 4. \tag{26}$$

Based on (3), (4) and (8), $\beta_i(t)$ can be rewritten in terms of computed and measurable terms as follows:

$$\beta_i = \frac{1}{n^{*\mathrm{T}}R_d^{\mathrm{T}}A^{-1}p_{di}}. \tag{27}$$

Motivated by (23) and the defn of the NF in Definition 1, the desired camera pose trajectory is designed as follows:

$$\dot{\Upsilon}_d = -K(\Upsilon_d)L_{\Upsilon_d}^{\mathrm{T}}\nabla\varphi, \tag{28}$$

where $K(\Upsilon_d) \triangleq k_1(L_{\Upsilon_d}^{\mathrm{T}}L_{\Upsilon_d})^{-1}$, $k_1 \in \mathbb{R}$ denotes a positive constant, and $\nabla\varphi(\bar{p}_d) \triangleq (\partial\varphi(\bar{p}_d)/\partial\bar{p}_d)^{\mathrm{T}} \in \mathbb{R}^8$ denotes the gradient vector of $\varphi(\bar{p}_d)$. The development of a particular image-space NF and its gradient are provided in Appendix C. After substituting (28) into (23), the desired image trajectory can be expressed as follows:

$$\dot{\bar{p}}_d = -L_{\Upsilon_d}K(\Upsilon_d)L_{\Upsilon_d}^{\mathrm{T}}\nabla\varphi, \tag{29}$$

where it is assumed that $\nabla\varphi(\bar{p}_d)$ is not a member of the null space of $L_{\Upsilon_d}^{\mathrm{T}}(\bar{p}_d)$. Based on (23) and (28), it is clear that the desired image trajectory generated by (29) will satisfy the rigid body constraints.

Remark 2. Based on comments in Chaumette (1998) and the current development, it seems that a remaining open problem is to develop a rigorous, theoretical, and general approach to ensure that $\nabla\varphi(\bar{p}_d)$ is not a member of the null space of $L_{\Upsilon_d}^{\mathrm{T}}(\bar{p}_d)$ (i.e., $\nabla\varphi(\bar{p}_d) \notin NS(L_{\Upsilon_d}^{\mathrm{T}}(\bar{p}_d))$ where $NS(\cdot)$ denotes the null space operator). However, since the approach in this paper is developed in terms of the desired image-space trajectory (and hence, is an off-line approach), a particular desired image trajectory can be chosen (e.g., by trial and error) a priori to ensure that $\nabla\varphi(\bar{p}_d) \notin NS(L_{\Upsilon_d}^{\mathrm{T}}(\bar{p}_d))$. Similar comments are provided in Chaumette (1998) and Mezouar and Chaumette (2002) that indicate that in practice this assumption can be readily satisfied for particular cases. Likewise, a particular desired image trajectory is also assumed to be a priori selected to ensure that $\Upsilon_d(t)$, $\dot{\Upsilon}_d(t) \in \mathscr{L}_\infty$ if $\bar{p}_d(t) \in \mathscr{D}$. Based on the structure of (20) and (21), the assumption that $\Upsilon_d(t)$, $\dot{\Upsilon}_d(t) \in \mathscr{L}_\infty$ if $\bar{p}_d(t) \in \mathscr{D}$ is considered mild in the sense that the only possible alternative case is if the camera could somehow be positioned at an infinite distance from the target while all four feature points remain visible.

Remark 3. It is clear that $K(\Upsilon_d)$ is positive definite if $L_{\Upsilon_d}(\bar{p}_d)$ is full rank. Similar to the statement in Remark 2, this assumption is readily satisfied for the proposed off-line path planner approach. Based on this assumption, $K(\Upsilon_d)$ satisfies the following inequalities:

$$\underline{k}\|\xi\|^2 \leqslant \xi^{\mathrm{T}}K(\Upsilon_d)\xi \leqslant \bar{k}(\cdot)\|\xi\|^2 \quad \forall\xi \in \mathbb{R}^6, \tag{30}$$

where $\underline{k} \in \mathbb{R}$ denotes a positive constant, and $\bar{k}(\cdot)$ denotes a positive, non-decreasing function.

3.3. Path planner analysis

Theorem 1. *Provided that the desired feature points can be a priori selected to ensure that* $\bar{p}_d(0) \in \mathscr{D}$ *and that* $\nabla\varphi(\bar{p}_d) \notin NS(L_{\Upsilon_d}^{\mathrm{T}}(\bar{p}_d))$, *then the desired image trajectory generated by* (29) *ensures that* $\bar{p}_d(t) \in \mathscr{D}$ *and* (29) *has the asymptotically stable equilibrium point* $\bar{p}^*$.

Proof. Let $V_1(\bar{p}_d) : \mathscr{D} \to \mathbb{R}$ denote a non-negative function defined as follows:

$$V_1(\bar{p}_d) \triangleq \varphi(\bar{p}_d). \tag{31}$$

After taking the time derivative of (31), the following expression can be obtained:

$$\dot{V}_1(\bar{p}_d(t)) = (\nabla\varphi)^{\mathrm{T}}\dot{\bar{p}}_d. \tag{32}$$

After substituting (29) into (32), the following expression can be obtained:

$$\dot{V}_1(\bar{p}_d(t)) = -(L_{\gamma_d}^{\mathrm{T}}\nabla\varphi)^{\mathrm{T}} K(\gamma_d) L_{\gamma_d}^{\mathrm{T}}\nabla\varphi. \tag{33}$$

Based on (30), $\dot{V}_1(\bar{p}_d(t))$ can be upper bounded as follows:

$$\dot{V}_1(\bar{p}_d(t)) \leqslant -\underline{k}\|L_{\gamma_d}^{\mathrm{T}}\nabla\varphi\|^2, \tag{34}$$

which clearly shows that $V_1(\bar{p}_d(t))$ is a non-increasing function in the sense that

$$V_1(\bar{p}_d(t)) \leqslant V_1(\bar{p}_d(0)). \tag{35}$$

From (31), (35), and the development in Appendix C, it is clear that, for any initial condition $\bar{p}_d(0) \in \mathscr{D}$, $\bar{p}_d(t) \in \mathscr{D} \forall t > 0$; therefore, $\mathscr{D}$ is a positively invariant set (Khalil, 2002). Let $E_1 \subset \mathscr{D}$ denote the following set: $E_1 \triangleq \{\bar{p}_d(t) | \dot{V}_1(\bar{p}_d) = 0\}$. Based on (33), it is clear that $\|L_{\gamma_d}^{\mathrm{T}}(\bar{p}_d)\nabla\varphi(\bar{p}_d)\| = 0$ in E_1; hence, from (28) and (29), it can be determined that $\|\dot{\gamma}_d(t)\| = \|\dot{\bar{p}}_d(t)\| = 0$ in E_1, and that E_1 is the largest invariant set. By invoking LaSalle's Theorem (Khalil, 2002), it can be determined that every solution $\bar{p}_d(t) \in \mathscr{D}$ approaches E_1 as $t \to \infty$, and hence, $\|L_{\gamma_d}^{\mathrm{T}}(\bar{p}_d)\nabla\varphi(\bar{p}_d)\| \to 0$. Since $\bar{p}_d(t)$ are chosen *a priori* via the off-line path planning routine in (29), the four feature points can be *a priori* selected to ensure that $\nabla\varphi(\bar{p}_d) \notin NS(L_{\gamma_d}^{\mathrm{T}}(\bar{p}_d))$. Provided $\nabla\varphi(\bar{p}_d) \notin NS(L_{\gamma_d}^{\mathrm{T}}(\bar{p}_d))$, then $\|L_{\gamma_d}^{\mathrm{T}}(\bar{p}_d)\nabla\varphi(\bar{p}_d)\| = 0$ implies that $\|\nabla\varphi(\bar{p}_d)\| = 0$. Based on development given in Appendix C, we can now show that $\nabla\varphi(\bar{p}_d(t)) \to 0$, and, hence, that $\bar{p}_d(t) \to \bar{p}^*$. □

4. Tracking control development

Based on Theorem 1, the desired camera pose trajectory can be generated from (28) to ensure that the camera moves along a path generated in the image space such that the desired object features remain visible (i.e., $\bar{p}_d(t) \in \mathscr{D}$). The objective in this section is to develop a controller so that the actual camera pose $\gamma(t)$ tracks the desired camera pose $\gamma_d(t)$ generated by (28), while also ensuring that the object features remain visible (i.e., $\bar{p}(t) \triangleq [u_1(t) \quad v_1(t) \quad \ldots \quad u_4(t) \quad v_4(t)]^{\mathrm{T}} \in \mathscr{D}$). To quantify this objective, a rotational tracking error, denoted by $e_\omega(t) \in \mathbb{R}^3$, is defined as

$$e_\omega \triangleq \Theta - \Theta_d, \tag{36}$$

and a translational tracking error, denoted by $e_v(t) \in \mathbb{R}^3$, is defined as follows:

$$e_v = p_{e1} - p_{ed1}. \tag{37}$$

4.1. Control development

After taking the time derivative of (36) and (37), the open-loop dynamics for $e_\omega(t)$ and $e_v(t)$ can be obtained as follows:

$$\dot{e}_\omega = -L_\omega\omega_c - \dot{\Theta}_d, \tag{38}$$

$$\dot{e}_v = -\frac{1}{z_1}A_{e1}v_c + A_{e1}[m_1]_\times\omega_c - \dot{p}_{ed1}, \tag{39}$$

where (14) was utilized. Based on the open-loop error systems in (38) and (39), $v_c(t)$ and $\omega_c(t)$ are designed as follows:

$$\omega_c \triangleq L_\omega^{-1}(K_\omega e_\omega - \dot{\Theta}_d), \tag{40}$$

$$v_c \triangleq \frac{1}{\alpha_1}A_{e1}^{-1}(K_v e_v - \hat{z}_1^*\dot{p}_{ed1}) + \frac{1}{\alpha_1}[m_1]_\times\omega_c\hat{z}_1^*, \tag{41}$$

where K_ω, $K_v \in \mathbb{R}^{3\times3}$ denote diagonal matrices of positive constant control gains, and $\hat{z}_1^*(t) \in \mathbb{R}$ denotes a parameter estimate for z_1^* that is designed as follows:

$$\dot{\hat{z}}_1^* \triangleq k_2 e_v^{\mathrm{T}}(A_{e1}[m_1]_\times\omega_c - \dot{p}_{ed1}), \tag{42}$$

where $k_2 \in \mathbb{R}$ denotes a positive constant adaptation gain. After substituting (40) and (41) into (38) and (39), the following closed-loop error systems can be developed:

$$\dot{e}_\omega = -K_\omega e_\omega, \tag{43}$$

$$z_1^*\dot{e}_v = -K_v e_v + (A_{e1}[m_1]_\times\omega_c - \dot{p}_{ed1})\tilde{z}_1^*, \tag{44}$$

where the parameter estimation error signal $\tilde{z}_1^*(t) \in \mathbb{R}$ is defined as follows:

$$\tilde{z}_1^* = z_1^* - \hat{z}_1^*. \tag{45}$$

4.2. Controller analysis

Theorem 2. *The controller introduced in* (40) and (41), *along with the adaptive update law defined in* (42), *ensures that the actual camera pose tracks the desired camera pose trajectory in the sense that*

$$\|e_\omega(t)\| \to 0, \quad \|e_v(t)\| \to 0 \quad \text{as } t \to \infty. \tag{46}$$

Proof. Let $V_2(t) \in \mathbb{R}$ denote a non-negative function defined as follows:

$$V_2 \triangleq \frac{1}{2}e_\omega^{\mathrm{T}}e_\omega + \frac{z_1^*}{2}e_v^{\mathrm{T}}e_v + \frac{1}{2k_2}\tilde{z}_1^{*2}. \tag{47}$$

After taking the time derivative of (47) and then substituting for the closed-loop error systems developed in (43) and (44), the following expression can be obtained:

$$\dot{V}_2 = -e_\omega^{\mathrm{T}}K_\omega e_\omega - e_v^{\mathrm{T}}K_v e_v + e_v^{\mathrm{T}}(A_{e1}[m_1]_\times\omega_c - \dot{p}_{ed1})\tilde{z}_1^* - \frac{1}{k_2}\tilde{z}_1^*\dot{\tilde{z}}_1^*, \tag{48}$$

where the time derivative of (45) was utilized. After substituting the adaptive update law designed in (42) into (48), the following expression can be obtained:

$$\dot{V}_2 = -e_\omega^{\mathrm{T}}K_\omega e_\omega - e_v^{\mathrm{T}}K_v e_v. \tag{49}$$

Based on (45), (47) and (49), it can be determined that $e_\omega(t)$, $e_v(t)$, $\tilde{z}_1^*(t)$, $\hat{z}_1^*(t) \in \mathscr{L}_\infty$ and that $e_\omega(t)$, $e_v(t) \in \mathscr{L}_2$. Based on the assumption that $\dot{\Theta}_d(t)$ is bounded (see Remark 2), the expressions given in (36), (40), and $L_\omega(t)$ in (16) can be used to conclude that $\omega_c(t) \in \mathscr{L}_\infty$. Since $e_v(t) \in \mathscr{L}_\infty$, (37), (12), (8),

and $A_{e1}(t)$ in (15) can be used to prove that $u_1(t)$, $v_1(t)$, $\alpha_1(t)$, $m_1(t)$, $A_{e1}(t) \in \mathcal{L}_\infty$. Based on the assumption that $\dot{p}_{ed1}(t)$ is bounded (see Remark 2), the expressions in (41), (42), and (44) can be used to conclude that $v_c(t)$, $\dot{\hat{z}}_1^*(t)$,$\dot{e}_v(t) \in \mathcal{L}_\infty$. Since $e_\omega(t) \in \mathcal{L}_\infty$, it is clear from (43) that $\dot{e}_\omega(t) \in \mathcal{L}_\infty$. 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 (Slotine & Li, 1991) can be used to prove the result given in (46). □

Remark 4. Based on the result provided in (46), it can be proven from the Euclidean reconstruction given in (5) and (6) that $R(t) \to R_d(t)$, $m_1(t) \to m_{d1}(t)$, and $z_1(t) \to z_{d1}(t)$ (and hence, $x_f(t) \to x_{fd}(t)$). Based on these results, (1) can be used to also prove that $\bar{m}_i(t) \to \bar{m}_{di}(t)$. Since $\Pi(\cdot)$ is a unique mapping, we can conclude that the desired camera pose converges to the goal camera pose based on the previous result that $\bar{p}_d(t) \to \bar{p}^*$ from Theorem 1. Based on the above analysis, $\bar{m}_i(t) \to \bar{m}^*$.

Remark 5. Based on (47) and (49), the following inequality can be obtained:

$$\begin{aligned} e_\omega^{\mathrm{T}} e_\omega + e_v^{\mathrm{T}} e_v &\leqslant 2 \max\left\{1, \frac{1}{z_1^*}\right\} V_2(t) \\ &\leqslant 2 \max\left\{1, \frac{1}{z_1^*}\right\} V_2(0), \end{aligned} \tag{50}$$

where

$$V_2(0) = \frac{1}{2} e_\omega^{\mathrm{T}}(0) e_\omega(0) + \frac{z_1^*}{2} e_v^{\mathrm{T}}(0) e_v(0) + \frac{1}{2k_2} \tilde{z}_1^{*2}(0).$$

From (11), (20), (36), (37), and the inequality in (50), the following inequality can be developed:

$$\|\Upsilon - \Upsilon_d\| \leqslant \sqrt{2 \max\left\{1, \frac{1}{z_1^*}\right\} V_2(0)}. \tag{51}$$

Based on (19), the following expression can be developed:

$$\bar{p} = \Pi(\Upsilon) - \Pi(\Upsilon_d) + \bar{p}_d. \tag{52}$$

After applying the mean-value theorem to (52), the following inequality can be obtained:

$$\|\bar{p}\| \leqslant \|L_{\Upsilon'}\| \|\Upsilon - \Upsilon_d\| + \|\bar{p}_d\|, \tag{53}$$

where every element of Υ' lies between the corresponding elements of Υ and Υ_d. Since all signals are bounded, it can be shown that $L_{\Upsilon'} \in \mathcal{L}_\infty$; hence, the following inequality can be developed from (51) and (53):

$$\|\bar{p}\| \leqslant \zeta_b \sqrt{V_2(0)} + \|\bar{p}_d\| \tag{54}$$

for some positive constant $\zeta_b \in \mathbb{R}$, where $\bar{p}_d(t) \in \mathcal{D}$ based on Theorem 1. To ensure that $\bar{p}(t) \in \mathcal{D}$, the image space needs to be sized to account for the effects of $\zeta_b V_2(0)$. Based on (47), $V_2(0)$ can be made arbitrarily small by increasing k_2 and initializing $\bar{p}_d(0)$ close or equal to $\bar{p}(0)$.

5. Simulation results

To solve the self-occlusion problem (the terminology, self-occlusion, in this paper is utilized to denote the case when the center of the camera is in the plane determined by the feature points) from a practical point of view, we define a distance ratio $\gamma(t) \in \mathbb{R}$ as follows:

$$\gamma(t) = \frac{d}{d^*}. \tag{55}$$

From Malis et al. (1999), $\gamma(t)$ is measurable. The idea to avoid the self-occlusion is to plan a desired image trajectory without self-occlusion. Based on (54), we can assume that the actual trajectory is close enough to the desired trajectory such that no self-occlusion occurred for the actual trajectory.

To illustrate the performance of the path planner given in (29) and the controller given in (40)–(42), numerical simulations will be performed for four standard visual servo tasks, which are believed to represent the most interesting tasks encountered by a visual servo system (Gans, Hutchinson, & Corke, 2003):

- Task 1: Optical axis rotation, a pure rotation about the optic axis.
- Task 2: Optical axis translation, a pure translation along the optic axis.
- Task 3: Camera y-axis rotation, a pure rotation of the camera about the y-axis of the camera coordinate frame.
- Task 4: General camera motion, a transformation that includes a translation and rotation about an arbitrary axis.

For the simulation, the intrinsic camera calibration matrix is given as follows:

$$A = \begin{bmatrix} f k_u & -f k_u \cot\phi & u_0 \\ 0 & \dfrac{f k_v}{\sin\phi} & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{56}$$

where $u_0 = 257$ and $v_0 = 253$ pixels represent the pixel coordinates of the principal point, $k_u = 101.4$ and $k_v = 101.4$ pixels mm^{-1} represent camera scaling factors, $\phi = 90°$ is the angle between the camera axes, and $f = 12.5$ mm denotes the camera focal length. For all simulations, we select $p_i(0) = p_{di}(0)$ $\forall i = 1, 2, 3, 4$, $\kappa = 8$.

5.1. Simulation results: optical axis rotation

The desired and actual image trajectories of the feature points are depicted in Figs. 2 and 3, respectively. The translational and rotational tracking errors of the target are depicted in Figs. 4 and 5, respectively, and the parameter estimate signal is depicted in Fig. 6. For the resulting Figs. 2–8, the control parameters were selected as follows:

$$K_v = I_3, \quad K_\omega = 0.3 I_3, \quad k_1 = 400,000, \quad k_2 = 0.04,$$

$$K = \mathrm{diag}\{10, 10, 10, 18, 13, 15, 10, 10\}.$$

The control input velocities $\omega_c(t)$ and $v_c(t)$ defined in (40) and (41) are depicted in Figs. 7 and 8. From Figs. 2 and 3, it is

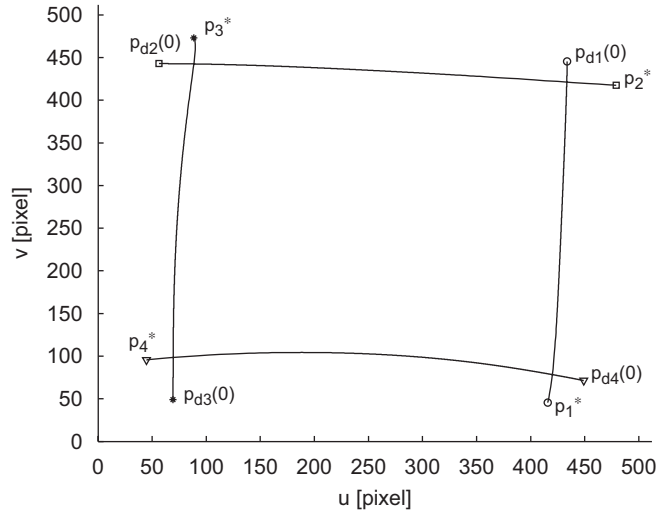

Fig. 2. Desired image trajectory of task 1.

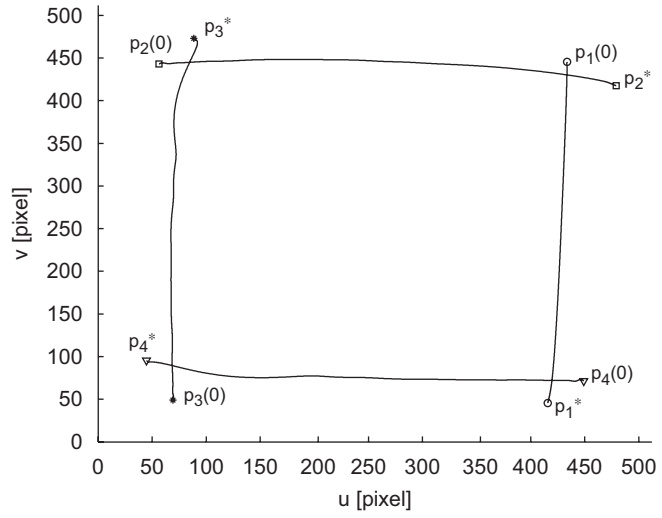

Fig. 3. Actual image trajectory of task 1.

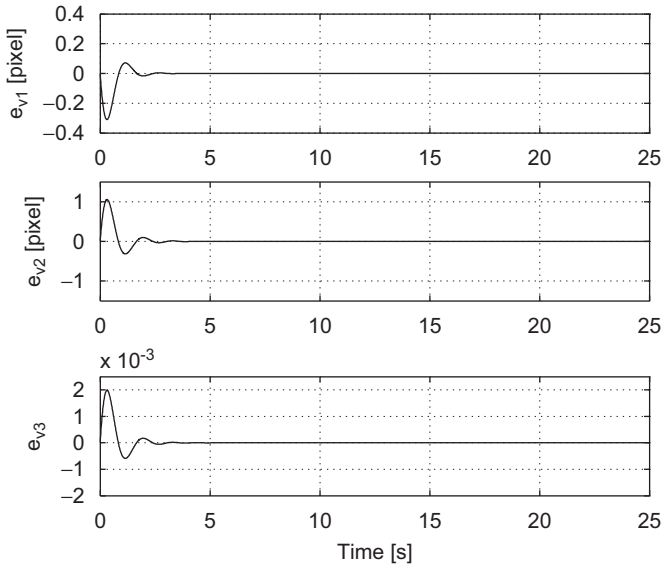

Fig. 4. Translational tracking error of task 1.

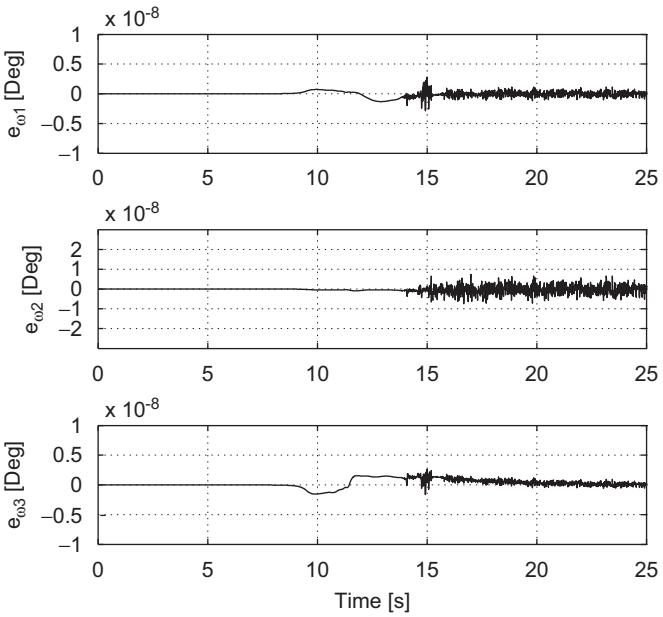

Fig. 5. Rotational tracking error of task 1.

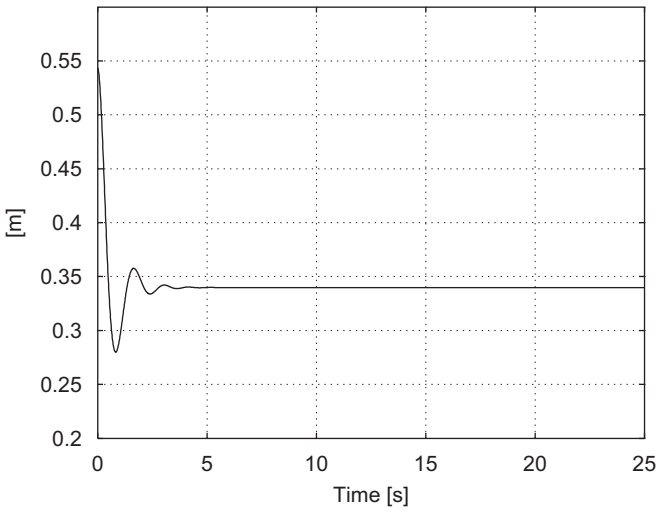

Fig. 6. Estimate of z_1^* of task 1.

clear that the desired feature points and actual feature points remain in the camera FOV and converge to the goal feature points. Figs. 4 and 5 show that the tracking errors go to zero as $t \to \infty$.

5.2. Simulation results: optical axis translation

The control parameters were selected as follows:

$$K_v = I_3, \quad K_\omega = 0.3I_3, \quad k_1 = 10,000 \quad k_2 = 0.0004,$$

$$K = \text{diag}\{30, 20, 10, 28, 33, 25, 10, 40\}.$$

The actual image trajectories of the feature points are depicted in Fig. 9. The control input velocities $\omega_c(t)$ and $v_c(t)$ defined

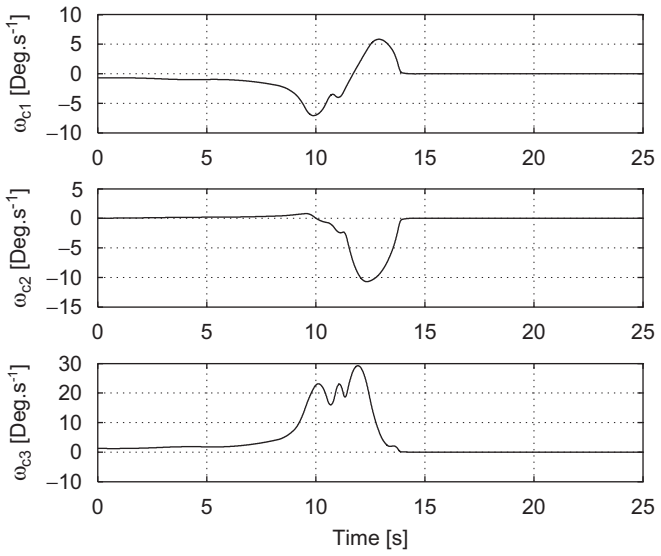

Fig. 7. Angular velocity of task 1.

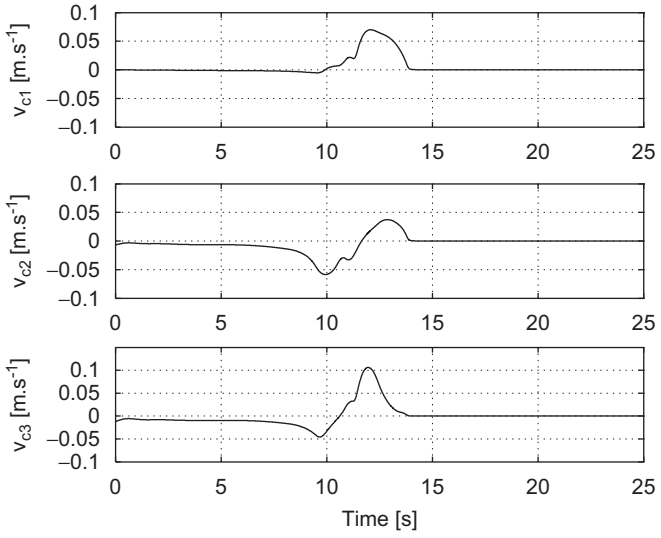

Fig. 8. Linear velocity input of task 1.

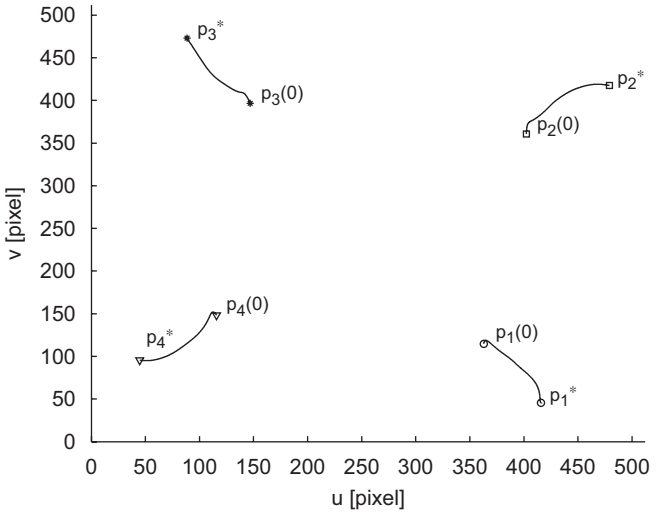

Fig. 9. Actual image trajectory of task 2.

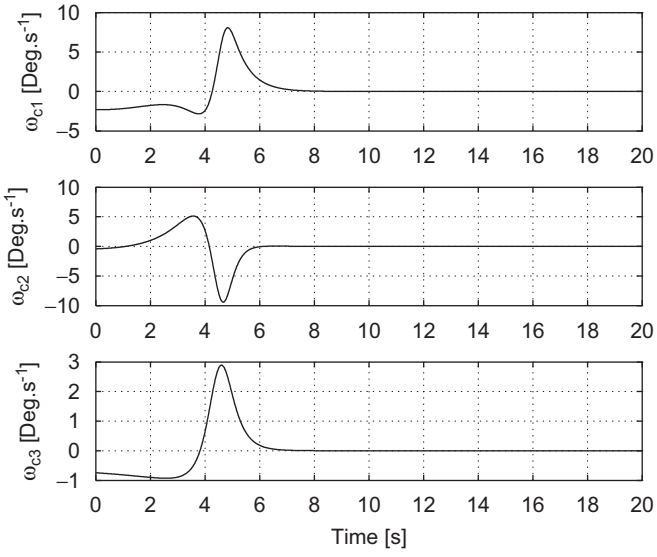

Fig. 10. Angular velocity input of task 2.

in (40) and (41) are depicted in Figs. 10 and 11. From Fig. 9, it is clear that the actual feature points remain in the camera FOV and converge to the goal feature points. To reduce the length of the paper, we only provide figures for the actual image trajectory, angular velocity, and linear velocity for this task (also for tasks 3 and 4). For more simulation results, see Chen (2005).

5.3. Simulation results: camera y-axis rotation

The control parameters were selected as follows:

$$K_v = 5I_3, \quad K_\omega = 0.3I_3, \quad k_1 = 1,000,000, \quad k_2 = 0.04,$$

$$K = \mathrm{diag}\{30, 20, 10, 28, 33, 25, 10, 40\}.$$

The actual image trajectories of the feature points are depicted in Fig. 12. The control input velocities $\omega_c(t)$ and $v_c(t)$ defined in (40) and (41) are depicted in Figs. 13 and 14. From Fig. 12, it is clear that the actual feature points remain in the camera FOV and converge to the goal feature points.

5.4. Simulation results: general camera motion

The control parameters were selected as follows:

$$K_v = I_3, \quad K_\omega = 0.3I_3, \quad k_1 = 200,000, \quad k_2 = 0.004,$$

$$K = \mathrm{diag}\{10, 10, 10, 18, 13, 15, 10, 10\}.$$

The actual image trajectories of the feature points are depicted in Fig. 15. The control input velocities $\omega_c(t)$ and $v_c(t)$ defined in (40) and (41) are depicted in Figs. 16 and 17. From Fig. 15,

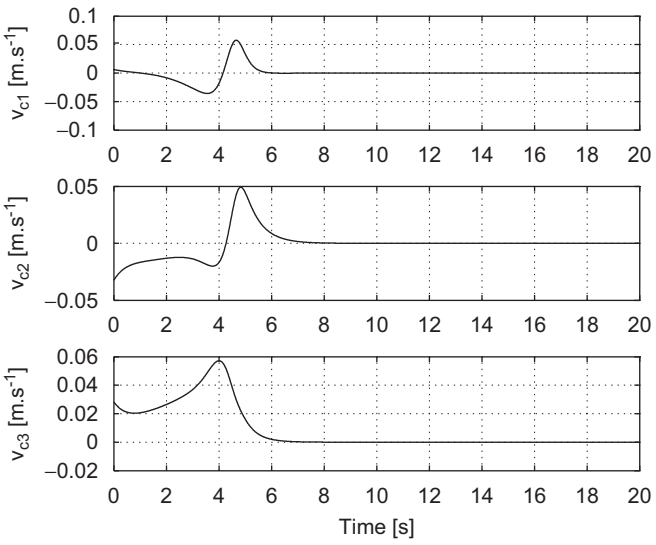

Fig. 11. Linear velocity input of task 2.

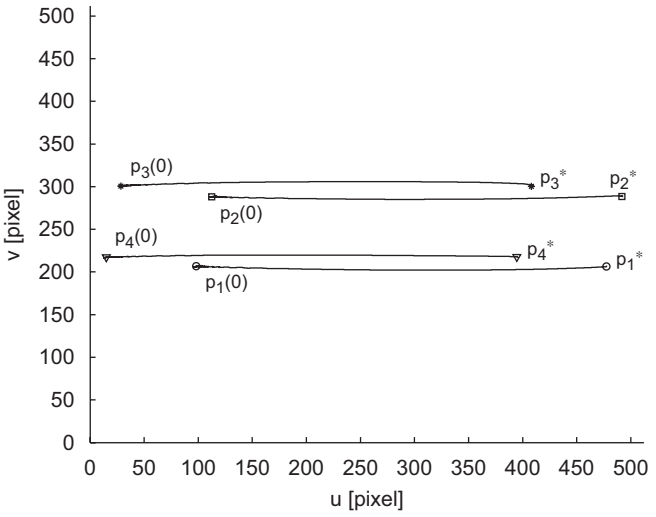

Fig. 12. Actual image trajectory of task 3.

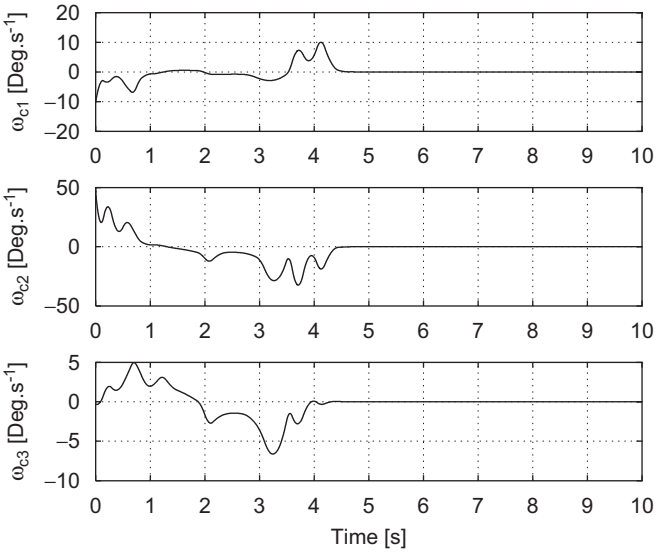

Fig. 13. Angular velocity input of task 3.

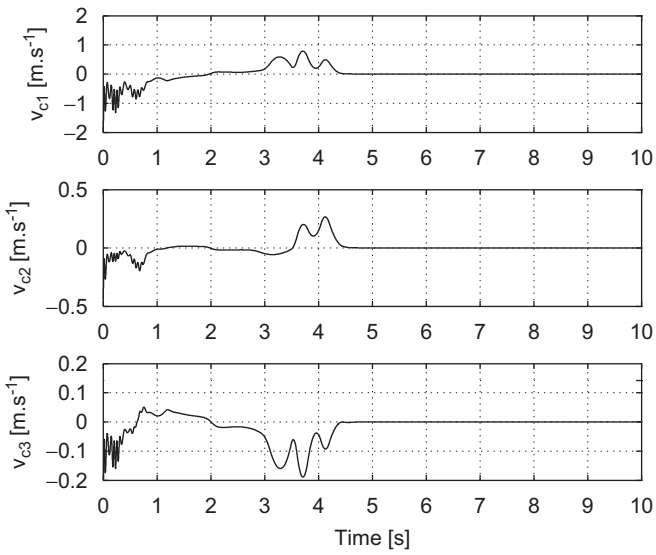

Fig. 14. Linear velocity input of task 3.

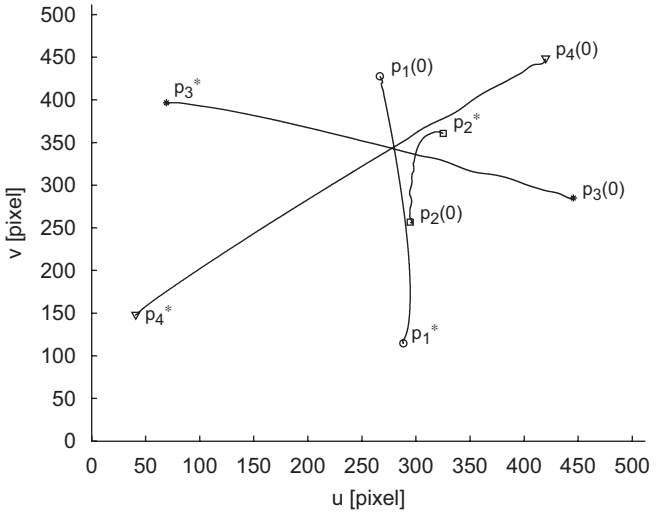

Fig. 15. Actual image trajectory of task 4.

it is clear that the actual feature points remain in the camera FOV and converge to the goal feature points.

Unlike position-based visual servoing and 2.5D visual servoing, our visual servo algorithm applies an image-based NF to ensure all feature points remain in the camera FOV. From Figs. 3, 9, 12, 15, it is clear that all actual image trajectories are very close to straight lines connecting the initial images and the goal images which is the main advantage of image-based visual servoing with regard to keeping the feature points in the camera FOV. Since our desired image trajectories are generated by (29), they satisfy the rigid body constraints. The common image-based visual servoing algorithms do not have mechanisms to satisfy rigid body constraints which cause IBVS robot systems to behave irregularly when the initial error is large. By monitoring the distance ratio $\gamma(t)$, all the desired trajectories in our simulations avoid the self-occlusion problem.

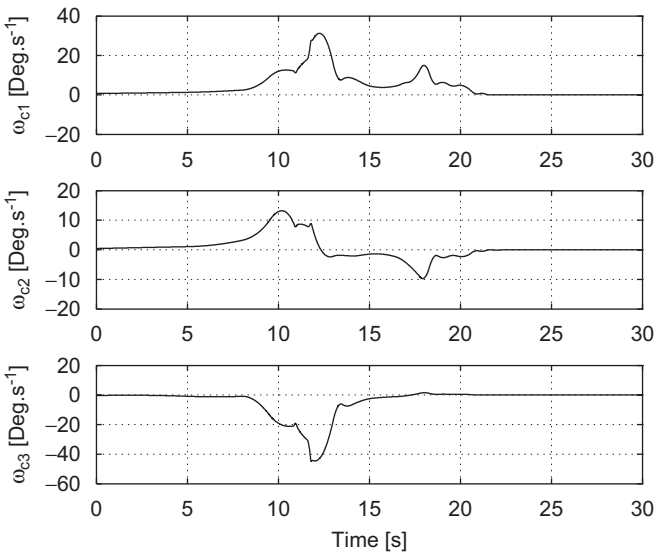

Fig. 16. Angular velocity input of task 4.

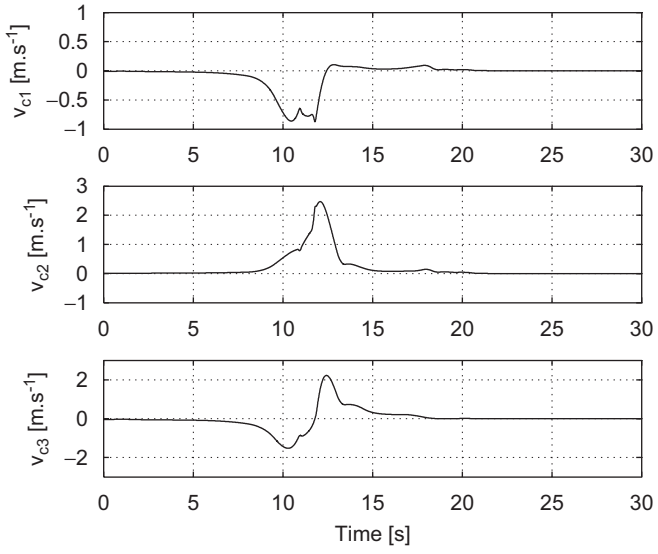

Fig. 17. Linear velocity input of task 4.

By comparing the desired image trajectories and the actual image trajectories in our simulations, it is clear that they are very close to each other. During closed-loop operation, we had verified that there is no self-occlusion for the actual robot system in the simulations. To the best of our knowledge, no other result can guarantee that all feature points remain in the camera FOV without irregular behavior while avoiding self-occlusion for full 6-DOF robot systems.

6. Conclusions

A path planner is developed based on an image-space NF that ensures the desired image trajectory converges to the goal position while also ensuring the desired image features remain in a visibility set under certain technical restrictions. An adaptive, homography-based visual servo tracking controller is then developed to navigate the camera-in-hand pose along the desired trajectory despite the lack of depth information from a monocular camera system. The path planner and the tracking controller are analyzed through a Lyapunov-based analysis. Simulation results are provided to illustrate the performance of the proposed approach. Further experimental verification of the performance of our proposed algorithm will be provided in the future when relevant facilities are available.

Appendix A. Open-loop dynamics

The extended image coordinates $p_{e1}(t)$ of (12) can be written as follows:

$$p_{e1} = \begin{bmatrix} a_1 & a_2 & 0 \\ 0 & a_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x_1}{z_1} \\ \frac{y_1}{z_1} \\ \ln(z_1) \end{bmatrix} + \begin{bmatrix} a_4 \\ a_5 \\ -\ln(z_1^*) \end{bmatrix}, \tag{A.1}$$

where (7), (8) and (9) were utilized. After taking the time derivative of (A.1), the following expression can be obtained:

$$\dot{p}_{e1} = \frac{1}{z_1} A_{e1} \dot{\bar{m}}_1.$$

By exploiting the fact that $\dot{\bar{m}}_1(t)$ can be expressed as follows:

$$\dot{\bar{m}}_1 = -v_c + [\bar{m}_1]_\times \omega_c,$$

the open-loop dynamics for $p_{e1}(t)$ can be rewritten as follows:

$$\dot{p}_{e1} = -\frac{1}{z_1} A_{e1} v_c + A_{e1} [m_1]_\times \omega_c.$$

The open-loop dynamics for $\Theta(t)$ can be expressed as follows (Chen et al., 2005):

$$\dot{\Theta} = -L_\omega \omega_c.$$

Appendix B. Image Jacobian-like matrix

Similar to (14), the dynamics for $\Upsilon_d(t)$ can be expressed as

$$\dot{\Upsilon}_d = \begin{bmatrix} \dot{p}_{ed1} \\ \dot{\Theta}_d \end{bmatrix} = \begin{bmatrix} -\frac{1}{z_{d1}} A_{ed1} & A_{ed1}[m_{d1}]_\times \\ 0_3 & -L_{\omega d} \end{bmatrix} \begin{bmatrix} v_{cd} \\ \omega_{cd} \end{bmatrix}, \tag{B.1}$$

where $\Theta_d(t)$ is defined in (22), $z_{di}(t)$ is introduced in (4), $A_{edi}(u_{di}, v_{di})$ is defined in the same manner as in (15) with respect to the desired pixel coordinates $u_{di}(t)$, $v_{di}(t)$, $m_{di}(t)$ is given in (1), $L_{\omega d}(\theta_d, \mu_d)$ is defined in the same manner as in (16) with respect to $\theta_d(t)$ and $\mu_d(t)$, and $v_{cd}(t)$, $\omega_{cd}(t) \in \mathbb{R}^3$ denote the desired linear and angular velocity signals that ensure compatibility with (B.1). The signals $v_{cd}(t)$ and $\omega_{cd}(t)$ are not actually used in the trajectory generation scheme presented in this paper as similarly done in Chen et al. (2005); rather, these signals are simply used to clearly illustrate how $\dot{\bar{p}}_d(t)$ can be expressed in terms of $\dot{\Upsilon}_d(t)$ as required in (23). Specifically, we first note that the top block row in (B.1) can be

used to write the time derivative of $p_{ed2}(t)$ in terms of $v_{cd}(t)$ and $\omega_{cd}(t)$ with $i = 2$:

$$\dot{p}_{ed2} = \left[-\frac{1}{z_{d2}} A_{ed2} \quad A_{ed2}[m_{d2}]_\times \right] \begin{bmatrix} v_{cd} \\ \omega_{cd} \end{bmatrix}, \tag{B.2}$$

where $p_{edi}(t)$ is defined in the same manner as (21) $\forall i = 1, 2, 3, 4$. After inverting the relationship given by (B.1), we can also express $v_{cd}(t)$ and $\omega_{cd}(t)$ as a function of $\dot{\Upsilon}_d(t)$ as follows:

$$\begin{bmatrix} v_{cd} \\ \omega_{cd} \end{bmatrix} = \begin{bmatrix} -z_{d1} A_{ed1}^{-1} & -z_{d1}[m_{d1}]_\times L_{\omega d}^{-1} \\ 0 & -L_{\omega d}^{-1} \end{bmatrix} \dot{\Upsilon}_d. \tag{B.3}$$

After substituting (B.3) into (B.2), $\dot{p}_{ed2}(t)$ can be expressed in terms of $\dot{\Upsilon}_d(t)$ as follows:

$$\dot{p}_{ed2} = \left[\frac{z_{d1}}{z_{d2}} A_{ed2} A_{ed1}^{-1} \quad A_{ed2} \left[\frac{z_{d1}}{z_{d2}} m_{d1} - m_{d2} \right]_\times L_{\omega d}^{-1} \right] \dot{\Upsilon}_d. \tag{B.4}$$

After formulating similar expressions for $\dot{p}_{ed3}(t)$ and $\dot{p}_{ed4}(t)$ as the one given by (B.4) for $\dot{p}_{ed2}(t)$, we can compute the expression for $L_{\Upsilon_d}(\bar{p}_d)$ in (24) by utilizing the definitions of $p_{di}(t)$ and $p_{edi}(t)$ given in (7) and (21), respectively (i.e., we must eliminate the bottom row of the expression given by (B.4)).

Appendix C. Image-space NF

Inspired by the framework developed in Cowan et al. (2002), an image-space NF is constructed by developing a diffeomorphism[4] between the image space and a model space, developing a model space NF, and transforming the model space NF into an image-space NF through the diffeomorphism (since NFs are invariant under diffeomorphism Koditschek & Rimon, 1990). To this end, a diffeomorphism is defined that maps the desired image feature vector $\bar{p}_d$ to the auxiliary model space signal $\zeta(\bar{p}_d) \triangleq [\zeta_1(\bar{p}_d)\zeta_2(\bar{p}_d) \dots \zeta_8(\bar{p}_d)]^{\mathrm{T}} : [-1, 1]^8 \to \mathbb{R}^8$ as follows:

$$\zeta = \operatorname{diag}\left\{ \frac{2}{u_{\max} - u_{\min}}, \frac{2}{v_{\max} - v_{\min}}, \dots, \frac{2}{v_{\max} - v_{\min}} \right\} \bar{p}_d - \left[\frac{u_{\max} + u_{\min}}{u_{\max} - u_{\min}} \quad \frac{v_{\max} + v_{\min}}{v_{\max} - v_{\min}} \dots \frac{v_{\max} + v_{\min}}{v_{\max} - v_{\min}} \right]^{\mathrm{T}}. \tag{C.1}$$

In (C.1), $u_{\max}$, $u_{\min}$, $v_{\max}$, and $v_{\min} \in \mathbb{R}$ denote the maximum and minimum pixel values along the u- and v- axis, respectively. The model space NF, denoted by $\tilde{\varphi}(\zeta) \in \mathbb{R}^8 \to \mathbb{R}$, is defined as follows (Cowan et al., 2002):

$$\tilde{\varphi}(\zeta) \triangleq \frac{\bar{\varphi}}{1 + \bar{\varphi}}. \tag{C.2}$$

In (C.2), $\bar{\varphi}(\zeta) \in \mathbb{R}^8 \to \mathbb{R}$ is defined as

$$\bar{\varphi}(\zeta) \triangleq \frac{1}{2} f(\zeta)^{\mathrm{T}} K f(\zeta), \tag{C.3}$$

where the auxiliary function $f(\zeta) : (-1, 1)^8 \to \mathbb{R}^8$ is defined similar to (Cowan et al., 2002) as follows:

$$f(\zeta) = \left[\frac{\zeta_1 - \zeta_1^*}{(1 - \zeta_1^{2\kappa})^{1/2\kappa}} \dots \frac{\zeta_8 - \zeta_8^*}{(1 - \zeta_8^{2\kappa})^{1/2\kappa}} \right]^{\mathrm{T}}, \tag{C.4}$$

where $K \in \mathbb{R}^{8\times 8}$ is a positive definite, symmetric matrix, and κ is a positive parameter. The reason we use κ instead of 1 as in (Cowan et al., 2002) is to get an additional parameter to change the potential field formed by $f(\zeta)$. See (Cowan et al., 2002) for a proof that (C.2) satisfies the properties of an NF as described in Definition 1. The image-space NF, denoted by $\varphi(\bar{p}_d) \in \mathscr{D} \to \mathbb{R}$, can then be developed as follows:

$$\varphi(\bar{p}_d) \triangleq \tilde{\varphi} \circ \zeta(\bar{p}_d), \tag{C.5}$$

where $\circ$ denotes the composition operator. The gradient vector $\nabla\varphi(p_d)$ can be expressed as follows:

$$\nabla\varphi \triangleq \left(\frac{\partial \varphi}{\partial \bar{p}_d} \right)^{\mathrm{T}} = \left(\frac{\partial \tilde{\varphi}}{\partial \zeta} \frac{\partial \zeta}{\partial \bar{p}_d} \right)^{\mathrm{T}}. \tag{C.6}$$

In (C.6), the partial derivative expressions $\partial\zeta(\bar{p}_d)/\partial\bar{p}_d$, $\partial\tilde{\varphi}(\zeta)/\partial\zeta$, and $\partial f(\zeta)/\partial\zeta$ can be expressed as follows:

$$\frac{\partial \zeta}{\partial \bar{p}_d} = \operatorname{diag}\left\{ \frac{2}{u_{\max} - u_{\min}}, \frac{2}{v_{\max} - v_{\min}}, \dots, \frac{2}{v_{\max} - v_{\min}} \right\}, \tag{C.7}$$

$$\frac{\partial \tilde{\varphi}}{\partial \zeta} = \frac{1}{(1 + \bar{\varphi})^2} f^{\mathrm{T}} K \frac{\partial f}{\partial \zeta}, \tag{C.8}$$

$$\frac{\partial f}{\partial \zeta} = \operatorname{diag}\left\{ \frac{1 - \zeta_1^{2\kappa-1}\zeta_1^*}{(1 - \zeta_1^{2\kappa})^{(2\kappa+1)/2\kappa}}, \dots, \frac{1 - \zeta_8^{2\kappa-1}\zeta_8^*}{(1 - \zeta_8^{2\kappa})^{(2\kappa+1)/2\kappa}} \right\}. \tag{C.9}$$

It is clear from (C.1)–(C.9) that $\bar{p}_d(t) \to \bar{p}^*$ when $\nabla\varphi(\bar{p}_d) \to 0$.

[4] A diffeomorphism is a map between manifolds which is differentiable and has a differentiable inverse.

References

Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, & A. Morse (Eds.), *The confluence of vision and control. Lecture notes in control and information science series* (Vol. 237, pp. 66–78). New York: Springer.

Chaumette, F., Malis, E., & Boudet, S. (1997). 2D 1/2 visual servoing with respect to a planar object. In *Proceedings of the workshop on new trends in image-based robot servoing* (pp. 45–52).

Chen, J. (2005). *Visual servo control with a monocular camera*. Ph.D. dissertation, Department of Electrical and Computer Engineering, Clemson University, Clemson, SC.

Chen, J., Dawson, D. M., Dixon, W. E., & Behal, A. (2005). Adaptive homography-based visual servo tracking for fixed camera configuration with a camera-in-hand extension. *IEEE Transactions on Control System Technology*, *13*(5), 814–825.

Chen, J., Dixon, W. E., Dawson, D. M., & McIntire, M. (2006). Homography-based visual servo tracking control of a wheeled mobile robot. *IEEE Transactions on Robotics*, *22*(2), 407–416.

Corke, P. I., & Hutchinson, S. A. (2000). A new hybrid image-based visual servo control scheme. In *Proceedings of the IEEE conference on decision and control* (pp. 2521–2527), Las Vegas, NV, December.

Cowan, N. J., Shakernia, O., Vidal, R., & Sastry, S. (2003). Vision-based follow-the-leader. In *Proceedings of the international conference on intelligent robots and systems* (pp. 1796–1801), Las Vegas, NV, October.

Cowan, N. J., Weingarten, J. D., & Koditscheck, D. E. (2002). Visual servoing via navigation function. *IEEE Transactions on Robotics and Automation*, *18*(4), 521–533.

Deguchi, K. (1998). Optimal motion control for image-based visual servoing by decoupling translation and rotation. In *Proceedings of the international conference on intelligent robots and systems* (pp. 705–711), Victoria, BC, Canada, October.

Faugeras, O. (2001). *Three-dimensional computer vision*. Cambridge, MA: The MIT Press.

Fierro, R., Song, P., Das, A., & Kumar, V. (2002). Cooperative control of robot formations. In *Cooperative control and optimization* (Vol. 66, pp. 73–93). Dordrecht: Kluwer Academic Press.

Gans, N. R., & Hutchinson, S. A. (2003). An asymptotically stable switched system visual controller for eye in hand robots. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 735–742), Las Vegas, NV, October.

Gans, N. R., Hutchinson, S. A., & Corke, P. I. (2003). Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control. *International Journal of Robotics Research*, *22*(10–11), 955–981.

Hager, G. D., & Hutchinson, S. (guest editors). (1996). Special section on vision-based control of robot manipulators. *IEEE Transactions on Robotics and Automation 12*(5).

Horaud, R. (1987). New methods for matching 3-d objects with single perspective view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-9*(3), 401–412.

Khalil, H. K. (2002). *Nonlinear systems*. (3rd ed.), Englewood Cliffs, NJ: Prentice-Hall.

Koditschek, D. E., & Rimon, E. (1990). Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, *11*, 412–442.

Malis, E., & Chaumette, F. (2000). 2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, *37*(1), 79–97.

Malis, E., Chaumette, F., & Boudet, S. (1999). 2 1/2 D visual servoing. *IEEE Transactions on Robotics and Automation*, *15*(2), 238–250.

Mezouar, Y., & Chaumette, F. (2002). Path planning for robust image-based control. *IEEE Transactions on Robotics and Automation*, *18*(4), 534–549.

Pereira, G. A., Das, A. K., Kumar, V., & Campos, M. F. (2003). Formation control with configuration space constraints. In *Proceedings of the international conference on intelligent robots and systems* (pp. 2755–2760), Las Vegas, NV, October.

Rimon, E., & Koditschek, D. E. (1992). Exact robot navigation using artificial potential function. *IEEE Transactions on Robotics and Automation*, *8*(5), 501–518.

Slotine, J. J. E., & Li, W. (1991). *Applied nonlinear control*. Englewood Cliffs, NJ: Prentice-Hall.

Spong, M. W., & Vidyasagar, M. (1989). *Robot dynamic and control*. New York, NY: Wiley.

Jian Chen received a B.E. degree in Testing Technology and Instrumentation, an M.E. degree in Control Science and Engineering, both from Zhejiang University, Hangzhou, PR China, in 1998 and 2001, respectively, and a Ph.D. degree in Electrical Engineering from Clemson University, Clemson, South Carolina, in 2005. After completing his Ph.D. program in August of 2005, he had worked on MEMS as a research associate at Clarkson University for 1 year. In August of 2006, he joined the University of Michigan, Ann Arbor, as a research fellow. His research interests include fuel cell modeling and control, MEMS, visual servo techniques, nonlinear control, and multi-vehicle navigation.

Darren M. Dawson received a B.S. degree in Electrical Engineering from the Georgia Institute of Technology in 1984. He then worked for Westinghouse as a control engineer from 1985 to 1987. In 1987, he returned to the Georgia Institute of Technology where he received the Ph.D. degree in Electrical Engineering in March 1990. In July 1990, he joined the Electrical and Computer Engineering Department where he currently holds the position of McQueen Quattlebaum Professor. His research interests are Nonlinear control techniques for mechatronic applications such as electric machinery, robotic systems, aerospace systems, acoustic noise, underactuated systems, magnetic bearings, mechanical friction, paper handling/textile machines, flexible beams/robots/rotors, cable structures, and vision-based systems. He also focuses on the development of Realtime hardware and software systems for control implementation.

Warren Dixon received his Ph.D. degree in 2000 from the Department of Electrical and Computer Engineering from Clemson University. After completing his doctoral studies he was selected as an Eugene P. Wigner Fellow at Oak Ridge National Laboratory (ORNL) where he worked in the Robotics and Energetic Systems Group. In 2004, he joined the faculty of the University of Florida in the Mechanical and Aerospace Engineering Department. His main research interest has been the development and application of Lyapunov-based control techniques for mechatronic systems, and he has published 2 books and over 100 refereed journal and conference papers. Recent publications have focused on Lyapunov-based control of nonlinear systems. He was awarded the 2001 ORNL Early Career Award for Engineering Achievement for his contributions to Lyapunov-based control methods. He was awarded the 2004 DOE Outstanding Mentor Award for his student advising at ORNL. He was awarded an NSF CAREER Award in 2006 for new development and application of Lyapunov-based control methods. He was also awarded the 2006 IEEE Robotics and Automation Society (RAS) Early Academic Career Award. He serves on the IEEE CSS Technical Committees on Intelligent Control and Nonlinear Systems, is a member of the ASME DSC Division Mechatronics Technical Committee, and is currently an associate editor for IEEE Transactions on Systems, Man, and Cybernetics; Part B: Cybernetics.

Dr. Vilas Kumar Chitrakaran received a B.E from the University of Madras, India in 1999, an M.S. from Clemson University, USA in 2003 and a Ph.D. from Clemson University in 2006, USA, all in Electrical Engineering. His Ph.D. dissertation focussed on vision-based nonlinear estimation and control strategies using Lyapunov design methods. He also has experience in designing real-time robotic and machine vision systems. Dr Chitrakaran is currently employed as a Systems Engineer in OC Robotics (Bristol, UK) where he is engaged in developing vision systems for snake-like continuum robots.