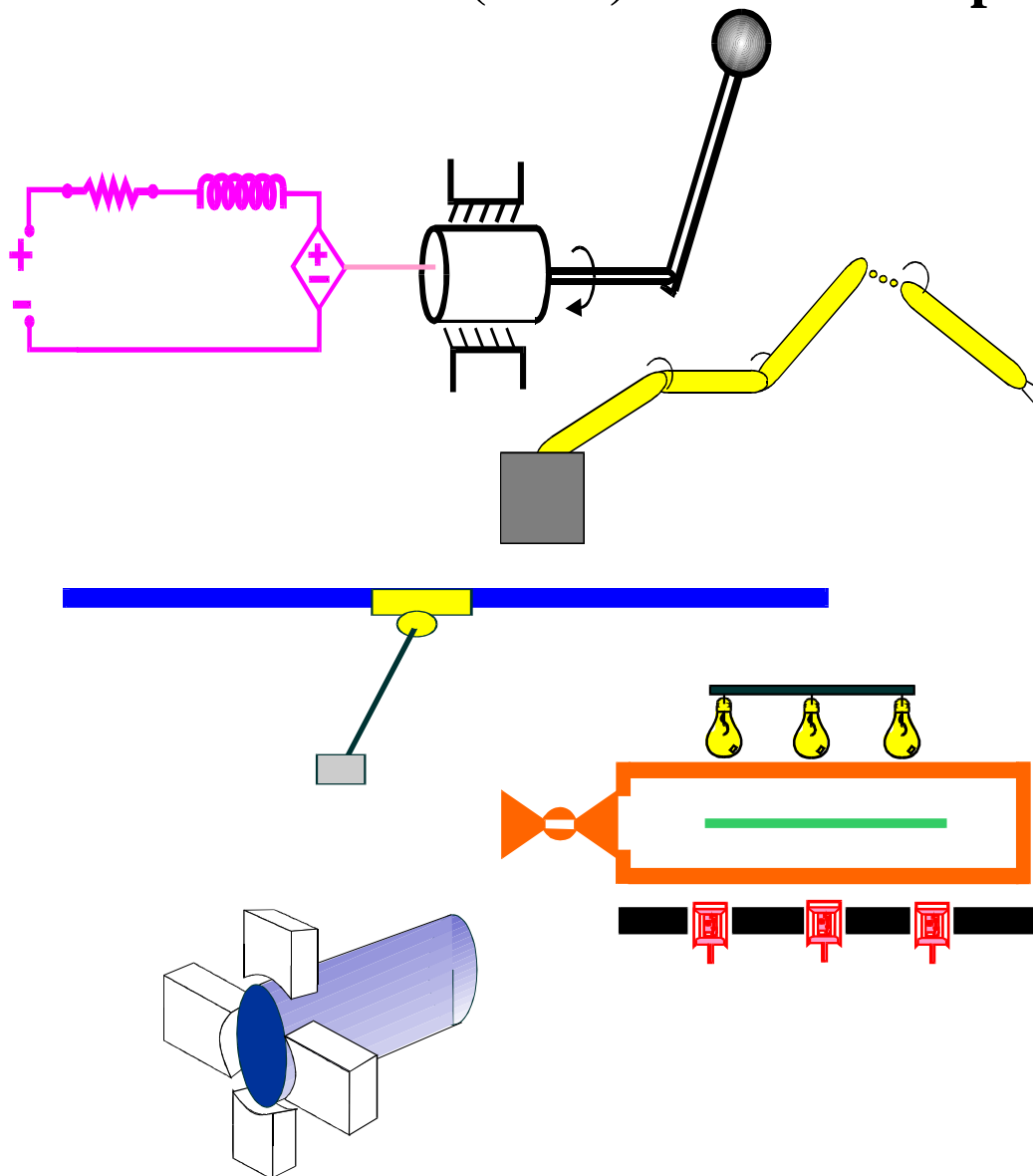


Clemson University
College of Engineering and Science
Control and Robotics (CRB) Technical Report



Number: CU/CRB/9/9/04/#1

Title: Navigation Function Based Visual Servo Control
Authors: J. Chen, D. M. Dawson, W. E. Dixon, and V. K.
Chitrakaran

Navigation Function Based Visual Servo Control¹

J. Chen,[†] D. M. Dawson,[†] W. E. Dixon,[‡] and V. K. Chitrakaran[†]

[†]Department of Electrical & Computer Engineering, Clemson University, Clemson, SC 29634-0915

[‡]Department of Mechanical & Aerospace Engineering, University of Florida, Gainesville, FL 32611

email: jianc@ces.clemson.edu, ddawson@ces.clemson.edu, wdixon@ufl.edu, cvilas@ces.clemson.edu

Abstract: *In this paper, a unique camera 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. The self-occlusion problem is also discussed. 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.*

1 Introduction

There is significant motivation to provide improved autonomy for robotic systems. In part, this motivation has lead 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 with associated the transformation between the image space and the Euclidean-space [2]. For a review of IBVS and PBVS controllers see [19].

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., [1], [25], [26]). The homography-based approach exploits a combination of reconstructed Euclidean 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 [25], 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 [5], [7], and [10]).

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 [28] where the path to a goal point is generated via a potential function that incorporates motion constraints; however, as stated in [28], local minima associated with traditional potential functions may exist.

¹This research was supported in part by U.S. NSF Grant DMI-9457967, ONR Grant N00014-99-1-0589, a DOC Grant, and an ARO Automotive Center Grant at Clemson University, and in part by AFOSR contract number F49620-03-1-0381 at the University of Florida.

Using a specialized potential function (coined a navigation function (NF)) originally proposed in [23] and [33], 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 [9]. However, as stated in [28], this approach requires the complete knowledge of the space topology and requires an object model. In [17], 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 [8] 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” [15]. Without a feedforward component, the controller in [8] yields an approximate “input-to-formation” stability (i.e., a local, linear exponential system with a bounded disturbance). A NF based approach to the follow-the-leader problem for a group of fully actuated holonomic mobile robots is considered in [30] 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 [30] to ensure that the NF decreases to the goal position, the stability of the overall system is not examined.

Motivated by the image space navigation function developed in [9], 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 [2]. 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. Further more, the desired image generated with this image Jacobian-like matrix will satisfy rigid body constraints automatically (The terminology, rigid body constraints, in this paper is utilized to denote the constraints for the image feature vector that the feature points have a fixed relative position to each other in Euclidean space). Building on our recent research in [5], an adaptive homography based visual tracking con-

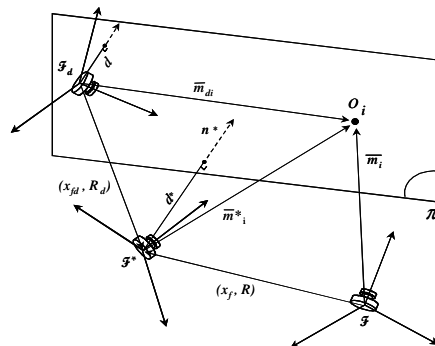


Figure 1: Coordinate frame relationships

troller 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 Figure 1), and are considered to be coplanar¹ 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

$$\begin{aligned} \bar{m}_i &= x_f + R\bar{m}_i^* \\ \bar{m}_{di} &= x_{fd} + R_d\bar{m}_i^* \end{aligned} \quad (1)$$

where $\bar{m}_i(t)$, $\bar{m}_{di}(t)$, and \bar{m}_i^* denote the Euclidean coordinates 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

¹It should be noted that if four coplanar target points are not available then the subsequent development can exploit the classic eight-points algorithm [26] with no four of the eight target points being coplanar.

\mathcal{F} to \mathcal{F}^* and \mathcal{F}_d to \mathcal{F}^* expressed in the coordinates of \mathcal{F} and \mathcal{F}_d , respectively. 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

$$m_i \triangleq \frac{\bar{m}_i}{z_i} \quad m_{di} \triangleq \frac{\bar{m}_{di}}{z_{di}} \quad m_i^* \triangleq \frac{\bar{m}_i^*}{z_i^*} \quad (2)$$

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), the expression in (1) can be rewritten as follows

$$m_i = \underbrace{\frac{z_i^*}{z_i}}_{\alpha_i} \underbrace{\left(R + \frac{x_f}{d^*} n^{*T} \right)}_H m_i^* \quad (3)$$

$$m_{di} = \underbrace{\frac{z_i^*}{z_{di}}}_{\alpha_{di}} \underbrace{\left(R_d + \frac{x_{fd}}{d^*} n^{*T} \right)}_{H_d} m_i^*. \quad (4)$$

In (3) and (4), $\alpha_i(t)$, $\alpha_{di}(t) \in \mathbb{R}$ denote invertible depth ratios, $H(t)$, $H_d(t) \in \mathbb{R}^{3 \times 3}$ denote Euclidean homographies [13], and $d^* \in \mathbb{R}$ denotes the constant, unknown distance from the origin of \mathcal{F}^* to π . The following projective relationship can also be developed from Fig. 1

$$d^* = n^{*T} \bar{m}_i^*. \quad (5)$$

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

$$d = n^{*T} R_d^T \bar{m}_{di}. \quad (6)$$

2.2 Projective Homography

Each feature point on π has a projected pixel coordinate 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

$$p_i \triangleq \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T \quad p_{di} \triangleq \begin{bmatrix} u_{di} & v_{di} & 1 \end{bmatrix}^T \\ p_i^* \triangleq \begin{bmatrix} u_i^* & v_i^* & 1 \end{bmatrix}^T. \quad (7)$$

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 (3) and (4) 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]

$$p_i = A m_i \quad p_{di} = A m_{di} \quad p_i^* = A m_i^* \quad (8)$$

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

$$A = \begin{bmatrix} a_1 & a_2 & a_4 \\ 0 & a_3 & a_5 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

where $a_i \in \mathbb{R} \forall i = 1, 2, \dots, 5$, denote known, constant calibration parameters (see [13]). After substituting (8) into (3) and (4), the following relationships can be developed

$$p_i = \underbrace{\alpha_i (A H A^{-1})}_G p_i^* \quad p_{di} = \underbrace{\alpha_{di} (A H_d A^{-1})}_{G_d} p_i^* \quad (10)$$

where $G(t)$, $G_d(t) \in \mathbb{R}^{3 \times 3}$ denote projective homographies. Given the images of the 4 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 [13]) can be used to compute $\alpha_i(t)$, $\alpha_{di}(t)$, n^* , $R(t)$, and $R_d(t)$ (see [5] 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

$$\Upsilon(t) \triangleq \begin{bmatrix} p_{e1}^T & \Theta^T \end{bmatrix}^T \quad (11)$$

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

$$p_{e1} = \begin{bmatrix} u_1 & v_1 & -\ln(\alpha_1) \end{bmatrix}^T, \quad (12)$$

and $\Theta(t) \in \mathbb{R}^3$ denotes the following axis-angle representation of $R(t)$ [32]

$$\Theta = \mu(t)\theta(t). \quad (13)$$

In (12), $\ln(\cdot)$ denotes the natural logarithm, and $\alpha_1(t)$ is introduced in (3). 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

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

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} ,

²The initial best-guess of n^* can be utilized to resolve the decomposition ambiguity. See [6] for details.

$A_{ei}(u_i, v_i) \in \mathbb{R}^{3 \times 3}$ is a known, invertible matrix defined as follows

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

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

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

where

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

Remark 1 As stated in [32], 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 solution³ for $\theta(t)$ and $\mu(t)$ can be determined as follows [32]

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

where the notation $\text{tr}(\cdot)$ denotes the trace of a matrix, and $[\mu_p]_\times$ denotes the 3×3 skew-symmetric expansion of $\mu_p(t)$. From (17), it is clear that

$$0 \leq \theta_p(t) \leq \pi. \quad (18)$$

3 Image-Based Path Planning

The path planning objective is to navigate the pose of a camera held by the end-effector of a robot manipulator to a desired camera pose along an image-space trajectory that ensures 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 $\bar{p}_d(t) \triangleq [u_{d1}(t) \ v_{d1}(t) \ \dots \ u_{d4}(t) \ v_{d4}(t)]^T \in \mathbb{R}^8$, 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 $\bar{p}^* \triangleq [u_1^* \ v_1^* \ \dots \ u_4^* \ v_4^*]^T \in \mathbb{R}^8$ is assumed to 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 a navigation function in [23], can be used. Specifically, the navigation functions used in this paper are defined as follows [33].

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

³See [4] for further details.

P 1) Analytic on \mathcal{D} (at least the first and second partial derivatives exist and are bounded on \mathcal{D});

P 2) a unique minimum exists at \bar{p}^* ;

P 3) it obtains a maximum value on the boundary of \mathcal{D} (i.e., admissible on \mathcal{D});

P 4) it is a Morse function (i.e., the matrix of second partial derivatives, the Hessian, evaluated at its critical points is nonsingular (and has bounded elements based on the smoothness property in P 1)).

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

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

where $\Pi(\cdot) : \mathbb{R}^6 \rightarrow \mathcal{D}$ denotes an unknown function mapping the camera pose to the image feature vector⁴. In (19), the desired camera pose is defined as follows

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

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

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

where $\alpha_{d1}(t)$ is introduced in (4), and $\Theta_d(t) \in \mathbb{R}^3$ denotes the axis-angle representation of $R_d(t)$ as follows

$$\Theta_d = \mu_d(t)\theta_d(t) \quad (22)$$

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 $R_d(t)$.

3.2 Desired Image Trajectory Planning

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

$$\dot{\bar{p}}_d = L_{\Upsilon_d} \dot{\Upsilon}_d \quad (23)$$

where $L_{\Upsilon_d}(\bar{p}_d) \triangleq \frac{\partial \bar{p}_d}{\partial \Upsilon_d} \in \mathbb{R}^{8 \times 6}$ 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

$$L_{\Upsilon_d} = \bar{I}T \quad (24)$$

⁴The 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 [20]. A unique camera pose can theoretically be obtained by using at least four points [2]. Therefore, the map $\Pi(\cdot)$ is a unique mapping with the image feature vector corresponding to a valid camera pose.

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 $I_n \in \mathbb{R}^{n \times n}$ denotes the $n \times n$ identity matrix, $0_n \in \mathbb{R}^{n \times n}$ 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

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

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 $\theta_d(t)$, and the auxiliary variable $\beta_i(t) \in \mathbb{R}$ is defined as follows

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

Based on (2), (6), and (8), $\beta_i(t)$ can be rewritten in terms of computed and measurable terms as follows

$$\beta_i = \frac{1}{n^* R_d^T A^{-1} p_{di}}. \quad (27)$$

Motivated by (23) and the definition of the navigation function in Definition 1, the desired camera pose trajectory is designed as follows

$$\dot{\Upsilon}_d = -k_1 L_{\Upsilon_d}^T \nabla \varphi \quad (28)$$

where $k_1 \in \mathbb{R}$ denotes a positive constant, and $\nabla \varphi(\bar{p}_d) \triangleq \left(\frac{\partial \varphi(\bar{p}_d)}{\partial \bar{p}_d} \right)^T \in \mathbb{R}^8$ 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

$$\dot{\bar{p}}_d = -k_1 L_{\Upsilon_d}^T \nabla \varphi \quad (29)$$

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

Remark 2 *Based on comments in [2] 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 $L_{\Upsilon_d}^T(\bar{p}_d)$ (i.e., $\nabla \varphi(\bar{p}_d) \notin NS(L_{\Upsilon_d}^T(\bar{p}_d))$ 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 $\nabla \varphi(\bar{p}_d) \notin NS(L_{\Upsilon_d}^T(\bar{p}_d))$. Similar comments are provided in [2] and [28] 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 $\Upsilon_d(t)$, $\dot{\Upsilon}_d(t) \in \mathcal{L}_\infty$ if $\bar{p}_d(t) \in \mathcal{D}$. Based on the structure of (20) and (21), the assumption that $\Upsilon_d(t)$, $\dot{\Upsilon}_d(t) \in \mathcal{L}_\infty$ if $\bar{p}_d(t) \in \mathcal{D}$ 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.

3.3 Path Planner Analysis

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

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

$$V_1(\bar{p}_d) \triangleq \varphi(\bar{p}_d). \quad (30)$$

After taking the time derivative of (30), the following expression can be obtained

$$\dot{V}_1(\bar{p}_d) = (\nabla \varphi)^T \dot{\bar{p}}_d. \quad (31)$$

After substituting (29) into (31) to obtain the following expression

$$\dot{V}_1(\bar{p}_d) = -k_1 \|L_{\Upsilon_d}^T \nabla \varphi\|^2, \quad (32)$$

it is clear that $V_1(\bar{p}_d)$ is a non-increasing function in the sense that

$$V_1(\bar{p}_d) \leq V_1(\bar{p}_d(0)). \quad (33)$$

From (30), (33), and the development in Appendix C, it is clear that for any initial condition $\bar{p}_d(0) \in \mathcal{D}$, that $\bar{p}_d(t) \in \mathcal{D} \forall t > 0$; therefore, \mathcal{D} is a positively invariant set [21]. Let $E_1 \subset \mathcal{D}$ denote the following set $E_1 \triangleq \{\bar{p}_d(t) \mid V_1(\bar{p}_d) = 0\}$. Based on (32), it is clear that $\|L_{\Upsilon_d}^T(\bar{p}_d) \nabla \varphi(\bar{p}_d)\| = 0$ in E_1 ; hence, from (28) and (29) it can be determined that $\|\dot{\Upsilon}_d(t)\| = \|\dot{\bar{p}}_d(t)\| = 0$ in E_1 , and that E_1 is the largest invariant set. By invoking LaSalle's Theorem [21], it can be determined that every solution $\bar{p}_d(t) \in \mathcal{D}$ approaches E_1 as $t \rightarrow \infty$, and hence, $\|L_{\Upsilon_d}^T(\bar{p}_d) \nabla \varphi(\bar{p}_d)\| \rightarrow 0$. 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 $\nabla\varphi(\bar{p}_d) \notin NS(L_{\Upsilon_d}^T(\bar{p}_d))$. Provided $\nabla\varphi(\bar{p}_d) \notin NS(L_{\Upsilon_d}^T(\bar{p}_d))$, then $\|L_{\Upsilon_d}^T(\bar{p}_d) \nabla\varphi(\bar{p}_d)\| = 0$ implies that $\|\nabla\varphi(\bar{p}_d)\| = 0$. Based on development of Appendix C, since $\nabla\varphi(\bar{p}_d) \rightarrow 0$ then $\bar{p}_d(t) \rightarrow \bar{p}^*$. \square

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., $\bar{p}_d(t) \in \mathcal{D}$). The objective in this section is to develop a controller so that the actual camera pose $\Upsilon(t)$ tracks the desired camera pose $\Upsilon_d(t)$ generated by (28), while also ensuring that the object features remain visible (i.e., $\bar{p}(t) \triangleq [u_1(t) \ v_1(t) \ \dots \ u_4(t) \ v_4(t)]^T \in \mathcal{D}$). To quantify this objective, a rotational tracking error, denoted by $e_\omega(t) \in \mathbb{R}^3$, is defined as

$$e_\omega \triangleq \Theta - \Theta_d, \quad (34)$$

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

$$e_v = p_{e1} - p_{ed1}. \quad (35)$$

4.1 Control Development

After taking the time derivative of (34) and (35), the open-loop dynamics for $e_\omega(t)$ and $e_v(t)$ can be obtained as follows

$$\dot{e}_\omega = -L_\omega \omega_c - \dot{\Theta}_d \quad (36)$$

$$\dot{e}_v = -\frac{1}{z_1} A_{e1} v_c + A_{e1} [m_1]_\times \omega_c - \dot{p}_{ed1} \quad (37)$$

where (14) was utilized. Based on the open-loop error systems in (36) and (37), $v_c(t)$ and $\omega_c(t)$ are designed as follows

$$\omega_c \triangleq L_\omega^{-1} (K_\omega e_\omega - \dot{\Theta}_d) \quad (38)$$

$$v_c \triangleq \frac{1}{\alpha_1} A_{e1}^{-1} (K_v e_v - \dot{z}_1^* \dot{p}_{ed1}) + \frac{1}{\alpha_1} [m_1]_\times \omega_c \dot{z}_1^* \quad (39)$$

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

$$\dot{\hat{z}}_1 \triangleq k_2 e_v^T (A_{e1} [m_1]_\times \omega_c - \dot{p}_{ed1}) \quad (40)$$

where $k_2 \in \mathbb{R}$ denotes a positive constant adaptation gain. After substituting (38) and (39) into (36) and (37), the following closed-loop error systems can be developed

$$\dot{e}_\omega = -K_\omega e_\omega \quad (41)$$

$$z_1^* \dot{e}_v = -K_v e_v + (A_{e1} [m_1]_\times \omega_c - \dot{p}_{ed1}) \dot{z}_1^* \quad (42)$$

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

$$\tilde{z}_1^* = z_1^* - \hat{z}_1^*. \quad (43)$$

4.2 Controller Analysis

Theorem 2 *The controller introduced in (38) and (39), along with the adaptive update law defined in (40), ensure that the actual camera pose tracks the desired camera pose trajectory in the sense that*

$$\|e_\omega(t)\| \rightarrow 0 \quad \|e_v(t)\| \rightarrow 0 \text{ as } t \rightarrow \infty. \quad (44)$$

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

$$V_2 \triangleq \frac{1}{2} e_\omega^T e_\omega + \frac{z_1^*}{2} e_v^T e_v + \frac{1}{2k_2} \tilde{z}_1^{*2}. \quad (45)$$

After taking the time derivative of (45) and then substituting for the closed-loop error systems developed in (41) and (42), the following expression can be obtained

$$\dot{V}_2 = -e_\omega^T K_\omega e_\omega - e_v^T K_v e_v + e_v^T (A_{e1} [m_1]_\times \omega_c - \dot{p}_{ed1}) \tilde{z}_1^* - \frac{1}{k_2} \tilde{z}_1^* \dot{z}_1^* \quad (46)$$

where the time derivative of (43) was utilized. After substituting the adaptive update law designed in (40) into (46), the following simplified expression can be obtained

$$\dot{V}_2 = -e_\omega^T K_\omega e_\omega - e_v^T K_v e_v. \quad (47)$$

Based on (43), (45), and (47), it can be determined that $e_\omega(t), e_v(t), \tilde{z}_1^*(t), \dot{z}_1^*(t) \in \mathcal{L}_\infty$ and that $e_\omega(t), e_v(t) \in \mathcal{L}_2$. Based on the assumption that $\dot{\Theta}_d(t)$ is bounded (see Remark 2), the expressions given in (34), (38), and $L_\omega(t)$ in (16) can be used to conclude that $\omega_c(t) \in \mathcal{L}_\infty$. Since $e_v(t) \in \mathcal{L}_\infty$, (35), (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 (39), (40), and (42) can be used to conclude that $v_c(t), \dot{z}_1^*(t), \dot{e}_v(t) \in \mathcal{L}_\infty$. Since $e_\omega(t) \in \mathcal{L}_\infty$, it is clear from (41) 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 [31] can be used to prove the result given in (44). \square

Remark 3 *Based on the result provided in (44), it can be proven from the Euclidean reconstruction given in (3) and (4) that $R(t) \rightarrow R_d(t)$, $m_1(t) \rightarrow m_{d1}(t)$, and $z_1(t) \rightarrow z_{d1}(t)$ (and hence, $x_f(t) \rightarrow x_{fd}(t)$). Based on these results, (1) can be used to also prove that $\bar{m}_i(t) \rightarrow \bar{m}_{di}(t)$. 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 $\bar{p}_d(t) \rightarrow \bar{p}^*$ from Theorem 1. Based on the above analysis, $\bar{m}_i(t) \rightarrow \bar{m}^*$.*

Remark 4 *Based on (45) and (47), the following inequality can be obtained*

$$e_\omega^T e_\omega + e_v^T e_v \leq 2 \max \left\{ 1, \frac{1}{z_1^*} \right\} V_2(t) \quad (48)$$

$$\leq 2 \max \left\{ 1, \frac{1}{z_1^*} \right\} V_2(0)$$

where

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

From (11), (20), (34), (35), and the inequality in (48), the following inequality can be developed

$$\|\Upsilon - \Upsilon_d\| \leq \sqrt{2 \max \left\{ 1, \frac{1}{z_1^*} \right\} V_2(0)}. \quad (49)$$

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

$$\bar{p} = \Pi(\Upsilon) - \Pi(\Upsilon_d) + \bar{p}_d. \quad (50)$$

After applying the mean-value theorem to (50), the following inequality can be obtained

$$\|\bar{p}\| \leq \|L_{\Upsilon_d}\| \|\Upsilon - \Upsilon_d\| + \|\bar{p}_d\|. \quad (51)$$

Since all signals are bounded, it can be shown that $L_{\Upsilon_d}^T(\bar{p}_d) \in \mathcal{L}_\infty$; hence, the following inequality can be developed from (49) and (51)

$$\|\bar{p}\| \leq \zeta_b \sqrt{V_2(0)} + \|\bar{p}_d\| \quad (52)$$

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 , $V_2(0)$ 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

From a practical point of view, we choose a state-related time varying control gain matrix $k_3 (L_{\Upsilon}^T L_{\Upsilon})^{-1}$ instead of a constant k_1 in (28) for the image path planner as follows

$$\dot{\Upsilon}_d = -k_3 (L_{\Upsilon}^T L_{\Upsilon})^{-1} L_{\Upsilon_d}^T \nabla \varphi \quad (53)$$

where $k_3 \in \mathbb{R}$ is a constant control gain. Through many simulation trials, we conclude that the path planner in (53) works better than the path planner in (28). Using the path planner in (53) instead of the path planner in (28) will not affect the proof for Theorem 1 as long as $L_{\Upsilon}^T L_{\Upsilon}$ is positive definite along the desired image trajectory $\bar{p}_d(t)$ (It is clear that $L_{\Upsilon}^T L_{\Upsilon}$ 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 this off-line path planner approach.

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

$$\gamma(t) = \frac{d}{d^*} \quad (54)$$

From [25], $\gamma(t)$ is measurable. The idea to avoid the self-occlusion is to plan a desired image trajectory without self-occlusion. Based on (52), we can assume that the actual trajectory is close enough to the desired trajectory such that no self-occlusion occurred for the actual trajectory. This assumption has been verified by many simulation trials.

To illustrate the performance of the path planner given in (53) and the controller given in (38)-(40), we implemented the simulations for four standard visual servo tasks, which are believed represent the most interesting tasks encountered by visual servo system [18]:

- 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

$$A = \begin{bmatrix} f k_u & -f k_u \cot \phi & u_0 \\ 0 & \frac{f k_v}{\sin \phi} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (55)$$

where $u_0 = 257$ [pixels], $v_0 = 253$ [pixels], $k_u = 101.4$ [pixels·mm⁻¹] and $k_v = 101.4$ [pixels·mm⁻¹] represent camera scaling factors, $\phi = 90$ [Deg] is the angle between the camera axes, and $f = 12.5$ [mm] denotes the camera focal length.

5.1 Simulation Results: Optical axis rotation

The initial desired image-space coordinates and the initial desired image-space coordinates of the 4 target points were selected as follows (in pixels)

$$\begin{aligned} p_1(0) &= p_{d1}(0) = [434 \quad 445 \quad 1]^T \\ p_2(0) &= p_{d2}(0) = [56 \quad 443 \quad 1]^T \\ p_3(0) &= p_{d3}(0) = [69 \quad 49 \quad 1]^T \\ p_4(0) &= p_{d4}(0) = [449 \quad 71 \quad 1]^T \end{aligned}$$

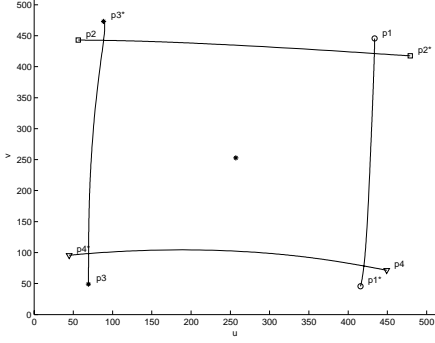


Figure 2: Desired Image Trajectory of Task 1

while the image-space coordinates of the 4 constant reference target points were selected as follows (in pixels)

$$\begin{aligned} p_1^* &= [416 \ 46 \ 1]^T & p_2^* &= [479 \ 418 \ 1]^T \\ p_3^* &= [88 \ 473 \ 1]^T & p_4^* &= [45 \ 96 \ 1]^T. \end{aligned} \quad (56)$$

The feedback gains K_v and K_ω , the adaptation gain k_2 and control gain k_3 , and the image navigation function parameter K and κ were adjusted through trial and error to the following values to yield improved performance

$$\begin{aligned} K_v &= \text{diag}\{1, 1, 1\} & K_\omega &= \text{diag}\{0.3, 0.3, 0.3\} \\ k_2 &= 0.04 & k_3 &= 400000 & \kappa &= 8 \\ K &= \text{diag}\{10, 10, 10, 18, 13, 15, 10, 10\}. \end{aligned}$$

The resulting desired image trajectory and actual image trajectory are depicted in Figure 2 and Figure 3, translational and rotational errors of the target are depicted in Figure 4 and Figure 5, respectively, and the parameter estimate signal is depicted in Figure 6. The control input velocities $\omega_c(t)$ and $v_c(t)$ defined in (38) and (39) are depicted in Figure 8 and 7. From Figure 2 and Figure 3, it is clear that the desired feature points and actual feature points remain in the camera field of view and converge to the goal feature points. Figure 4 and Figure 5 show that the tracking errors go to zero as $t \rightarrow \infty$. From Figure 4 to Figure 8, it is clear that all the signals are bounded.

5.2 Simulation Results: Optical axis translation

The initial desired image-space coordinates and the initial desired image-space coordinates of the 4 target points were selected as follows (in pixels)

$$\begin{aligned} p_1(0) = p_{d1}(0) &= [363 \ 115 \ 1]^T \\ p_2(0) = p_{d2}(0) &= [402 \ 361 \ 1]^T \\ p_3(0) = p_{d3}(0) &= [147 \ 397 \ 1]^T \\ p_4(0) = p_{d4}(0) &= [116 \ 148 \ 1]^T \end{aligned}$$

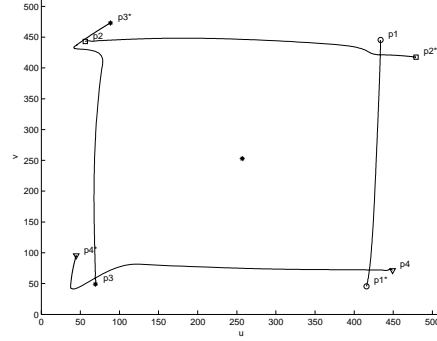


Figure 3: Actual Image Trajectory of Task 1

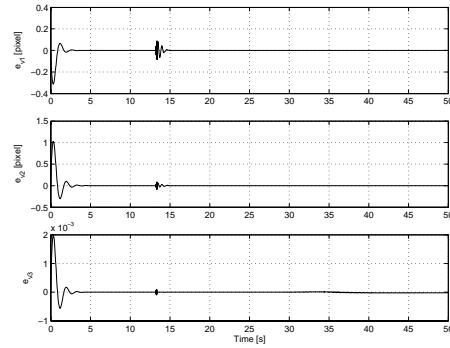


Figure 4: Translational Error of Task 1

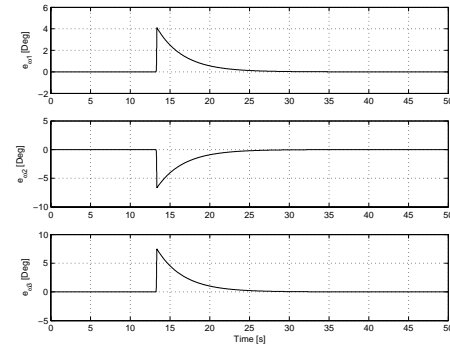


Figure 5: Rotational Error of Task 1

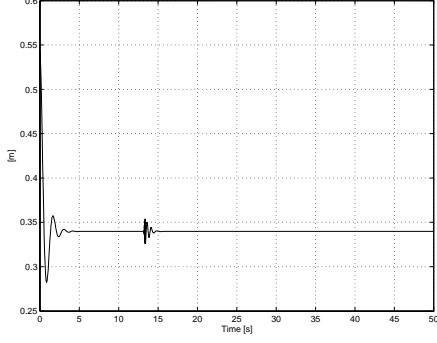


Figure 6: Estimate of z_1^* of Task 1

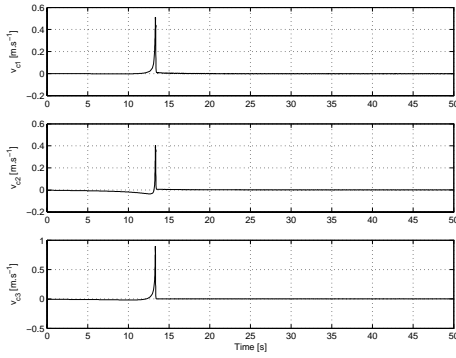


Figure 7: Linear Velocity Input of Task 1

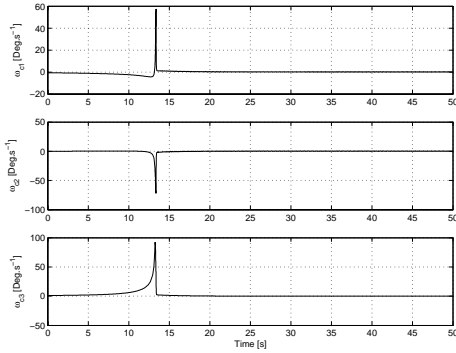


Figure 8: Angular Velocity of Task 1

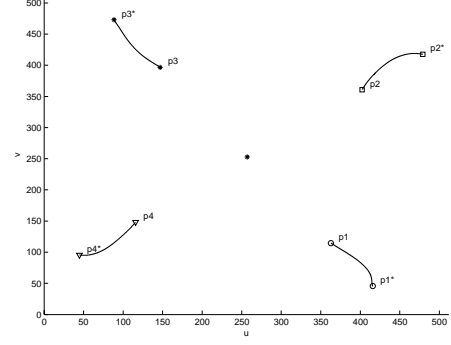


Figure 9: Desired Image Trajectory of Task 2

while the image-space coordinates of the 4 constant reference target points were selected as follows (in pixels)

$$\begin{aligned} p_1^* &= [416 \ 46 \ 1]^T & p_2^* &= [479 \ 418 \ 1]^T \\ p_3^* &= [88 \ 473 \ 1]^T & p_4^* &= [45 \ 96 \ 1]^T \end{aligned}$$

The feedback gains K_v and K_ω , the adaptation gain k_2 and control gain k_3 , and the image navigation function parameter K and κ were adjusted through trial and error to the following values to yield improved performance

$$K_v = \text{diag}\{1, 1, 1\} \quad K_\omega = \text{diag}\{0.3, 0.3, 0.3\}$$

$$k_2 = 0.0004 \quad k_3 = 10000 \quad \kappa = 8$$

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

The resulting desired image trajectory and actual image trajectory are depicted in Figure 9 and Figure 10, translational and rotational errors of the target are depicted in Figure 11 and Figure 12, respectively, and the parameter estimate signal is depicted in Figure 13. The control input velocities $\omega_c(t)$ and $v_c(t)$ defined in (38) and (39) are depicted in Figure 15 and 14. From Figure 9 and Figure 10, it is clear that the desired feature points and actual feature points remain in the camera field of view and converge to the goal feature points. Figure 11 and Figure 12 show that the tracking errors go to zero as $t \rightarrow \infty$. From Figure 11 to Figure 15, it is clear that all the signals are bounded.

5.3 Simulation Results: Camera y-axis rotation

The initial desired image-space coordinates and the initial desired image-space coordinates of the 4 target points were selected as follows (in pixels)

$$\begin{aligned} p_1(0) &= p_{d1}(0) = [98 \ 207 \ 1]^T \\ p_2(0) &= p_{d2}(0) = [112 \ 288 \ 1]^T \\ p_3(0) &= p_{d3}(0) = [29 \ 301 \ 1]^T \\ p_4(0) &= p_{d4}(0) = [15 \ 217 \ 1]^T \end{aligned}$$

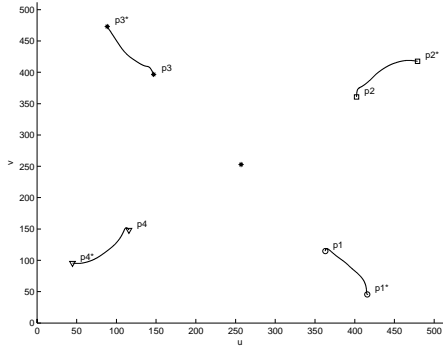


Figure 10: Actual Image Trajectory of Task 2

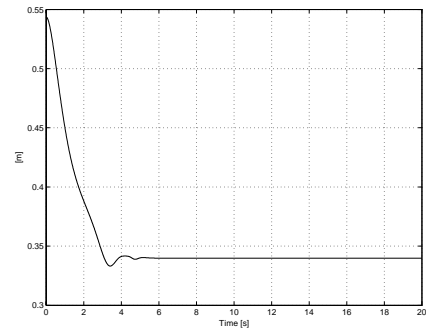


Figure 13: Estimate of z_1^* of Task 2

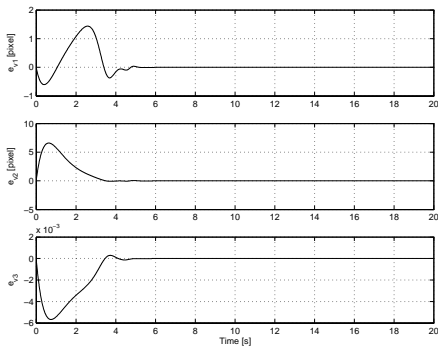


Figure 11: Translational Error of Task 2

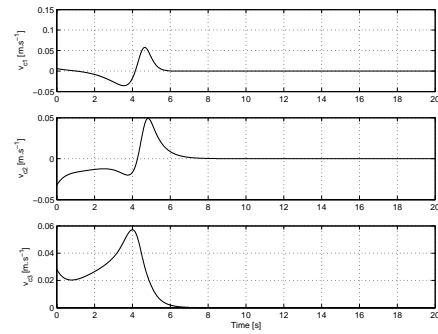


Figure 14: Linear Velocity Input of Task 2

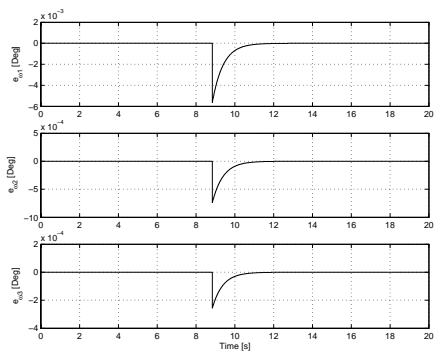


Figure 12: Rotational Error of Task 2

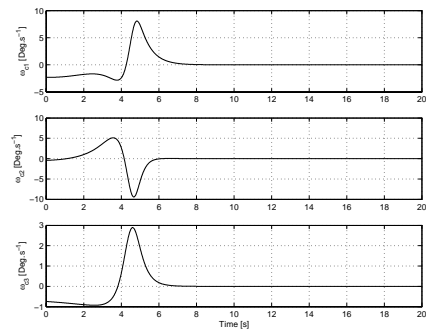


Figure 15: Angular Velocity Input of Task 2

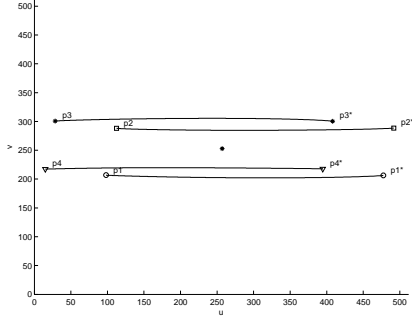


Figure 16: Desired Image Trajectory of Task 3

while the image-space coordinates of the 4 constant reference target points were selected as follows (in pixels)

$$\begin{aligned} p_1^* &= \begin{bmatrix} 478 & 206 & 1 \end{bmatrix}^T & p_2^* &= \begin{bmatrix} 492 & 289 & 1 \end{bmatrix}^T \\ p_3^* &= \begin{bmatrix} 408 & 300 & 1 \end{bmatrix}^T & p_4^* &= \begin{bmatrix} 395 & 218 & 1 \end{bmatrix}^T. \end{aligned}$$

The feedback gains K_v and K_ω , the adaptation gain k_2 and control gain k_3 , and the image navigation function parameter K and κ were adjusted through trial and error to the following values to yield improved performance

$$\begin{aligned} K_v &= \text{diag}\{1, 1, 1\} & K_\omega &= \text{diag}\{0.3, 0.3, 0.3\} \\ k_2 &= 0.0004 & k_3 &= 1000000 & \kappa &= 8 \\ K &= \text{diag}\{30, 20, 10, 28, 33, 25, 10, 40\}. \end{aligned}$$

The resulting desired image trajectory and actual image trajectory are depicted in Figure 16 and Figure 17, translational and rotational errors of the target are depicted in Figure 18 and Figure 19, respectively, and the parameter estimate signal is depicted in Figure 20. The control input velocities $\omega_c(t)$ and $v_c(t)$ defined in (38) and (39) are depicted in Figure 22 and 21. From Figure 16 and Figure 17, it is clear that the desired feature points and actual feature points remain in the camera field of view and converge to the goal feature points. Figure 18 and Figure 19 show that the tracking errors go to zero as $t \rightarrow \infty$. From Figure 18 to Figure 22, it is clear that all the signals are bounded.

5.4 Simulation Results: General Camera Motion

The initial desired image-space coordinates and the initial desired image-space coordinates of the 4 target points were selected as follows (in pixels)

$$\begin{aligned} p_1(0) &= p_{d1}(0) = \begin{bmatrix} 267 & 428 & 1 \end{bmatrix}^T \\ p_2(0) &= p_{d2}(0) = \begin{bmatrix} 295 & 257 & 1 \end{bmatrix}^T \\ p_3(0) &= p_{d3}(0) = \begin{bmatrix} 446 & 285 & 1 \end{bmatrix}^T \\ p_4(0) &= p_{d4}(0) = \begin{bmatrix} 420 & 449 & 1 \end{bmatrix}^T \end{aligned}$$

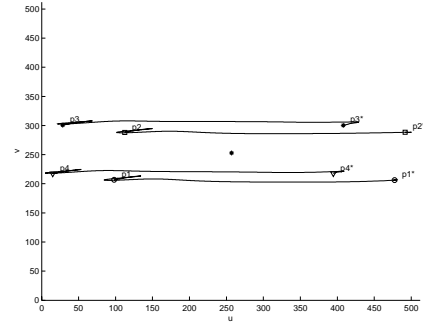


Figure 17: Actual Image Trajectory of Task 3

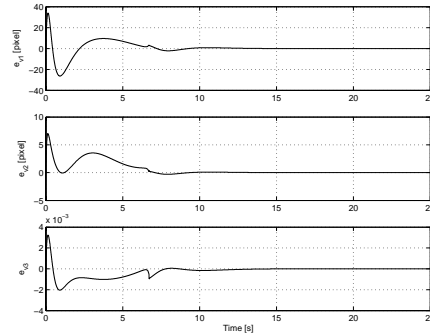


Figure 18: Translational Error of Task 3

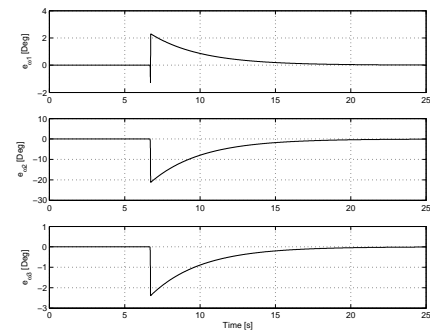


Figure 19: Rotational Error of Task 3

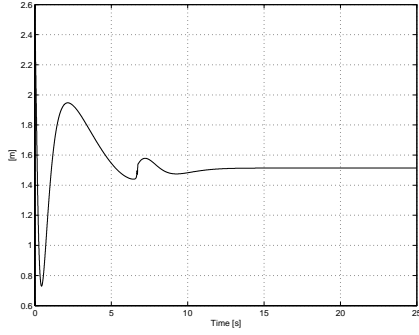


Figure 20: Estimate of z_1^* of Task 3

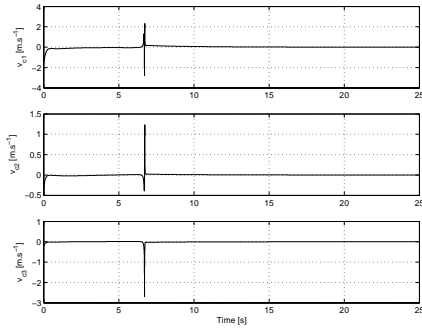


Figure 21: Linear Velocity Input of Task 3

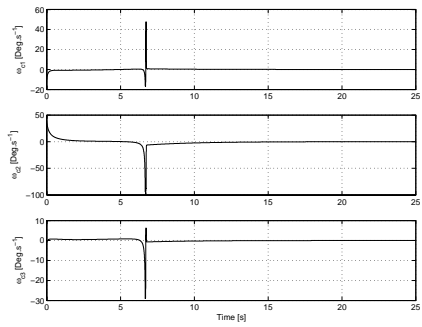


Figure 22: Angular Velocity Input of Task 3

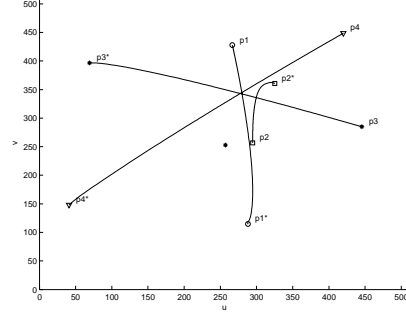


Figure 23: Desired Image Trajectory of Task 4

while the image-space coordinates of the 4 constant reference target points were selected as follows (in pixels)

$$\begin{aligned} p_1^* &= [416 \ 46 \ 1]^T & p_2^* &= [479 \ 418 \ 1]^T \\ p_3^* &= [88 \ 473 \ 1]^T & p_4^* &= [45 \ 96 \ 1]^T. \end{aligned}$$

The feedback gains K_v and K_ω , the adaptation gain k_2 and control gain k_3 , and the image navigation function parameter K and κ were adjusted through trial and error to the following values to yield improved performance

$$\begin{aligned} K_v &= \text{diag}\{1, 1, 1\} & K_\omega &= \text{diag}\{0.3, 0.3, 0.3\} \\ k_2 &= 0.004 & k_3 &= 200000 & \kappa &= 8 \\ K &= \text{diag}\{10, 10, 10, 18, 13, 15, 10, 10\}. \end{aligned}$$

The resulting desired image trajectory and actual image trajectory are depicted in Figure 23 and Figure 24, translational and rotational errors of the target are depicted in Figure 25 and Figure 26, respectively, and the parameter estimate signal is depicted in Figure 27. The control input velocities $\omega_c(t)$ and $v_c(t)$ defined in (38) and (39) are depicted in Figure 29 and 28. From Figure 23 and Figure 24, it is clear that the desired feature points and actual feature points remain in the camera field of view and converge to the goal feature points. Figure 25 and Figure 26 show that the tracking errors go to zero as $t \rightarrow \infty$. From Figure 25 to Figure 29, it is clear that all the signals are bounded.

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

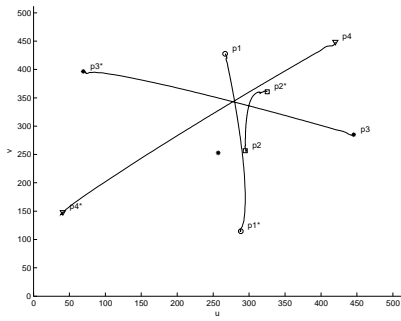


Figure 24: Actual Image Trajectory of Task 4

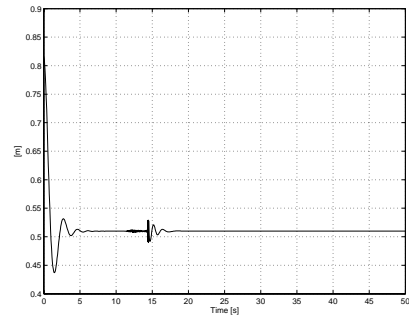


Figure 27: Estimate of z_1^* of Task 4

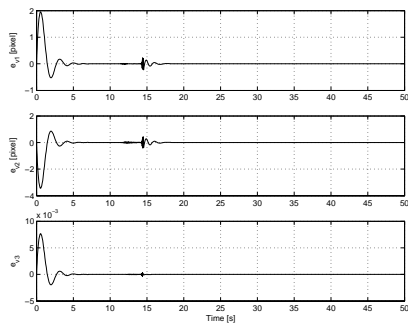


Figure 25: Translational Error of Task 4

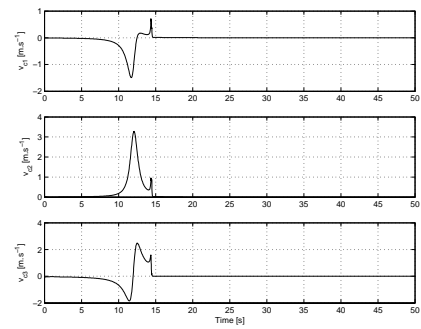


Figure 28: Linear Velocity Input of Task 4

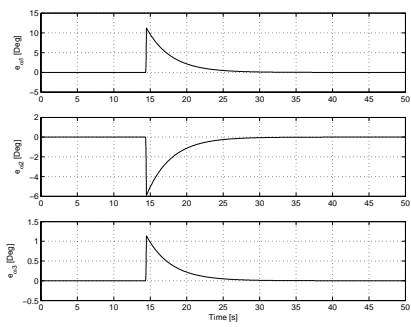


Figure 26: Rotational Error of Task 4

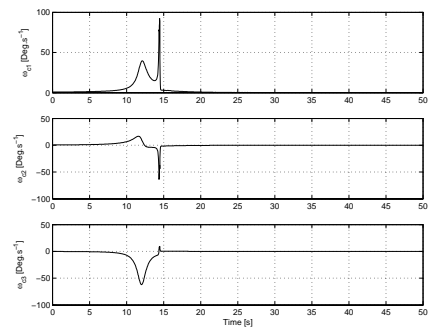


Figure 29: Angular Velocity Input of Task 4

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.

References

- [1] F. Chaumette, E. Malis, and S. Boudet, "2D 1/2 Visual Servoing with Respect to a Planar Object," Proc. of the Workshop on New Trends in Image-Based Robot Servoing, pp. 45-52, 1997.
- [2] F. Chaumette, "Potential Problems of Stability and Convergence in Image-Based and Position-Based Visual Servoing," The Confluence of Vision and Control, ser. LNCS, D. Kriegman, G. Hager, and A. Morse, Eds. New York: Springer Verlag, Vol. 237, pp. 66-78, 1998.
- [3] F. Chaumette and E. Malis, "2 1/2 D Visual Servoing: A Possible Solution to Improve Image-based and Position-based Visual Servoings," Proc. of the IEEE Intl. Conf. on Robotics and Automation, San Francisco, California, April 2000, pp. 630-635.
- [4] J. Chen, A. Behal, D. Dawson, and Y. Fang, "2.5D Visual Servoing with a Fixed Camera," Proc. of the American Control Conf., Denver, Colorado, June 2003, pp.3442-3447.
- [5] J. Chen, D. M. Dawson, W. E. Dixon, and A. Behal, "Adaptive Homography-based Visual Servo Tracking," Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, Las Vegas, Nevada, October 2003, pp. 230-235.
- [6] J. Chen, W. E. Dixon, D. M. Dawson, and M. McIntyre, "Homography-based Visual Servo Tracking Control of a Wheeled Mobile Robot," IEEE Transactions on Robotics, to appear (Sept. 2004).
- [7] P. I. Corke and S. A. Hutchinson, "A New Hybrid Image-Based Visual Servo Control Scheme," Proc. of the IEEE Conf. on Decision and Control, Las Vegas, Nevada, Dec. 2000, pp. 2521-2527.
- [8] N. J. Cowan, O. Shakernia, R. Vidal, and S. Sastry, "Vision-Based Follow-the-Leader," Proc. of the Intl. Conf. on Intelligent Robots and Systems, Las Vegas, Nevada, October 2003, pp. 1796-1801.
- [9] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek, "Visual Servoing via Navigation Function," IEEE Trans. on Robotics and Automation, Vol. 18, No. 4, pp. 521-533, (2002).
- [10] K. Deguchi, "Optimal Motion Control for Image-Based Visual Servoing by Decoupling Translation and Rotation," Proc. of the Intl. Conf. on Intelligent Robots and Systems, Victoria, B.C., Canada, Oct. 1998, pp. 705-711.
- [11] Y. Fang, Lyapunov-based Control for Mechanical and Vision-based System, Ph.D. dissertation, Dept. Elect. and Comput. Eng., Clemson Univ., Clemson, SC, 2002.
- [12] Y. Fang, W. E. Dixon, D. M. Dawson and J. Chen, "Robust 2.5D Visual Servoing for Robot Manipulators," Proc. of the American Control Conf., Denver, Colorado, June 2003, pp. 3311-3316.
- [13] O. Faugeras, Three-Dimensional Computer Vision, The MIT Press, Cambridge Massachusetts, 2001.
- [14] C. A. Felippa, A Systematic Approach to the Element-Independent Corotational Dynamics of Finite Elements, Center for Aerospace Structures Document Number CU-CAS-00-03, College of Engineering, University of Colorado, January 2000.
- [15] R. Fierro, P. Song, A. Das, and V. Kumar, "Cooperative Control of Robot Formations," In Cooperative Control and Optimization, Vol. 66, Chapter 5, pp. 73-93, Kluwer Academic Press, 2002.
- [16] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Communications ACM, Vol. 44, pp. 381-395, (1981).
- [17] N. R. Gans and S. A. Hutchinson, "An Asymptotically Stable Switched System Visual Controller for Eye in Hand Robots," Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, Las Vegas, Nevada, Oct. 2003, pp. 735-742.
- [18] N. R. Gans, S. A. Hutchinson, and P. I. Corke, "Performance Tests for Visual Servo Control Systems, with Application to Partitioned Approaches to Visual Servo Control," International Journal of Robotics Research, Vol. 22, No. 10-11, pp. 955-981, (2003).
- [19] G. D. Hager and S. Hutchinson (guest editors), Special Section on Vision-Based Control of Robot Manipulators, IEEE Trans. Robotics and Automation, Vol. 12, No. 5, (1996).
- [20] R. Horaud, "New Methods for Matching 3-D Objects with Single Perspective View," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 3, pp. 401-412, (1987).
- [21] H. K. Khalil, Nonlinear Systems, third edition, Prentice Hall, 2002.
- [22] D. E. Koditschek, "Exact Robot Navigation by Means of Potential Functions: Some Topological Considerations," Proc. of the IEEE Intl. Conf. on Robotics and Automation, Raleigh, North Carolina, May 1987, pp. 1-6.
- [23] D. E. Koditschek and E. Rimon, "Robot Navigation Functions on Manifolds with Boundary," Adv. Appl. Math., Vol. 11, pp. 412-442, (1990).
- [24] E. Malis, Contributions à la modélisation et à la commande en asservissement visuel, Ph.D. Dissertation, University of Rennes I, IRISA, France, Nov. 1998.
- [25] E. Malis, F. Chaumette, and S. Bodet, "2 1/2 D Visual Servoing," IEEE Trans. on Robotics and Automation, Vol. 15, No. 2, pp. 238-250, (1999).
- [26] E. Malis and F. Chaumette, "2 1/2 D Visual Servoing with Respect to Unknown Objects Through a New Estimation Scheme of Camera Displacement," Intl. Journal of Computer Vision, Vol. 37, No. 1, pp. 79-97, (2000).
- [27] E. Malis and F. Chaumette, "Theoretical Improvements in the Stability Analysis of a New Class of Model-Free Visual Servoing Methods," IEEE Trans. on Robotics and Automation, Vol. 18, No. 2, pp. 176-186, (2002).
- [28] Y. Mezouar and F. Chaumette, "Path Planning for Robust Image-Based Control," IEEE Trans. on Robotics and Automation, Vol. 18, No. 4, pp. 534-549, (2002).
- [29] Y. Mezouar and F. Chaumette, "Optimal Camera Trajectory with Image-Based Control," International Journal of Robotics Research, Vol. 22, No. 10-11, pp. 781-803, (2003).
- [30] G. A. Pereira, A. K. Das, V. Kumar, and M. F. Campos, "Formation Control with configuration Space Constraints," Proc. of the Intl. Conf. on Intelligent Robots and Systems, Las Vegas, Nevada, October 2003, pp. 2755-2760.
- [31] J. J. E. Slotine and W. Li, Applied Nonlinear Control, Prentice Hall, Inc: Englewood Cliff, NJ, 1991.

[32] M. W. Spong and M. Vidyasagar, Robot Dynamic and Control, John Wiley and Sons, Inc: New York, NY, 1989.

[33] E. Rimon and D. E. Koditschek, "Exact Robot Navigation Using Artificial Potential Function," IEEE Trans. on Robotics and Automation, Vol. 8, No. 5, pp. 501-518, (1992).

Appendix A

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

$$p_{e1} = \begin{bmatrix} a_1 & a_2 & 0 \\ 0 & a_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x_1}{z_1} \\ \frac{y_1}{z_1} \\ \ln(z_1) \end{bmatrix} + \begin{bmatrix} a_4 \\ a_5 \\ -\ln(z_1^*) \end{bmatrix} \quad (57)$$

where (7), (8), and (9) were utilized. After taking the time derivative of (57), the following expression can be obtained

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

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

$$\dot{m}_1 = -v_c + [m_1]_{\times} \omega_c,$$

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 [11]

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

Appendix B

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

$$\dot{\Upsilon}_d = \begin{bmatrix} \dot{p}_{ed1} \\ \dot{\Theta}_d \end{bmatrix} = \begin{bmatrix} -\frac{1}{z_{d1}} A_{ed1} & A_{ed1} [m_{d1}]_{\times} \\ 0_3 & -L_{\omega d} \end{bmatrix} \begin{bmatrix} v_{cd} \\ \omega_{cd} \end{bmatrix} \quad (58)$$

where $\Theta_d(t)$ is defined in (22), $z_{di}(t)$ is introduced in (2), $A_{edi}(u_{di}, v_{di})$ 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 (58). 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 [5]; 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 (58) can be used to write the time derivative of $p_{ed2}(t)$ in terms of $v_{cd}(t)$ and $\omega_{cd}(t)$ with $i = 2$

$$\dot{p}_{ed2} = \begin{bmatrix} -\frac{1}{z_{d2}} A_{ed2} & A_{ed2} [m_{d2}]_{\times} \end{bmatrix} \begin{bmatrix} v_{cd} \\ \omega_{cd} \end{bmatrix} \quad (59)$$

where $p_{edi}(t)$ is defined in the same manner as (21) $\forall i = 1, 2, 3, 4$. After inverting the relationship given by (58), we can also express $v_{cd}(t)$ and $\omega_{cd}(t)$ as a function of $\dot{\Upsilon}_d(t)$ as follows

$$\begin{bmatrix} v_{cd} \\ \omega_{cd} \end{bmatrix} = \begin{bmatrix} -z_{d1} A_{ed1}^{-1} & -z_{d1} [m_{d1}]_{\times} L_{\omega d}^{-1} \\ 0 & -L_{\omega d}^{-1} \end{bmatrix} \dot{\Upsilon}_d. \quad (60)$$

After substituting (60) into (59), $\dot{p}_{ed2}(t)$ can be expressed in terms of $\dot{\Upsilon}_d(t)$ as follows

$$\dot{p}_{ed2} = \begin{bmatrix} \frac{z_{d1}}{z_{d2}} A_{ed2} A_{ed1}^{-1} & A_{ed2} \left[\frac{z_{d1}}{z_{d2}} m_{d1} - m_{d2} \right]_{\times} L_{\omega d}^{-1} \end{bmatrix} \dot{\Upsilon}_d. \quad (61)$$

After formulating similar expressions for $\dot{p}_{ed3}(t)$ and $\dot{p}_{ed4}(t)$ as the one given by (61) 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 (61)).

Appendix C

Inspired by the framework developed in [9], an image space NF is constructed by developing a diffeomorphism⁵ 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 [23]). To this end, a diffeomorphism is defined that maps the desired image feature vector \bar{p}_d to the auxiliary model space signal $\zeta(\bar{p}_d) \triangleq [\zeta_1(\bar{p}_d) \zeta_2(\bar{p}_d) \dots \zeta_8(\bar{p}_d)]^T : [-1, 1]^8 \rightarrow \mathbb{R}^8$ as follows

$$\zeta = \text{diag} \left\{ \frac{2}{u_{\max} - u_{\min}}, \frac{2}{v_{\max} - v_{\min}}, \dots, \frac{2}{v_{\max} - v_{\min}} \right\} \bar{p}_d \quad (62)$$

$$- \begin{bmatrix} \frac{u_{\max} + u_{\min}}{u_{\max} - u_{\min}} & \frac{v_{\max} + v_{\min}}{v_{\max} - v_{\min}} & \dots & \frac{v_{\max} + v_{\min}}{v_{\max} - v_{\min}} \end{bmatrix}^T.$$

In (62), $u_{\max}, u_{\min}, v_{\max}$, and $v_{\min} \in \mathbb{R}$ denote the maximum and minimum pixel values along the u and v axes, respectively. The model space NF, denoted by $\tilde{\varphi}(\zeta) \in \mathbb{R}^8 \rightarrow \mathbb{R}$, is defined as follows [9]

$$\tilde{\varphi}(\zeta) \triangleq \frac{\bar{\varphi}}{1 + \bar{\varphi}}. \quad (63)$$

⁵A diffeomorphism is a map between manifolds which is differentiable and has a differentiable inverse.

In (63), $\bar{\varphi}(\zeta) \in \mathbb{R}^8 \rightarrow \mathbb{R}$ is defined as

$$\bar{\varphi}(\zeta) \triangleq \frac{1}{2} f(\zeta)^T K f(\zeta) \quad (64)$$

where the auxiliary function $f(\zeta) : (-1, 1)^8 \rightarrow \mathbb{R}^8$ is defined similar to [9] as follows

$$f(\zeta) = \left[\frac{\zeta_1 - \zeta_1^*}{(1 - \zeta_1^{2\kappa})^{1/2\kappa}} \quad \dots \quad \frac{\zeta_8 - \zeta_8^*}{(1 - \zeta_8^{2\kappa})^{1/2\kappa}} \right]^T \quad (65)$$

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 [9] is to get an additional parameter to change the potential field formed by $f(\zeta)$. See [9] for a proof that (63) satisfies the properties of a NF as described in Definition 1. The image space NF, denoted by $\varphi(\bar{p}_d) \in \mathcal{D} \rightarrow \mathbb{R}$, can then be developed as follows

$$\varphi(\bar{p}_d) \triangleq \tilde{\varphi} \circ \zeta(\bar{p}_d) \quad (66)$$

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

$$\nabla \varphi \triangleq \left(\frac{\partial \varphi}{\partial \bar{p}_d} \right)^T = \left(\frac{\partial \tilde{\varphi}}{\partial \zeta} \frac{\partial \zeta}{\partial \bar{p}_d} \right)^T. \quad (67)$$

In (67), the partial derivative expressions $\frac{\partial \zeta(\bar{p}_d)}{\partial \bar{p}_d}$, $\frac{\partial \tilde{\varphi}(\zeta)}{\partial \zeta}$, and $\frac{\partial f(\zeta)}{\partial \zeta}$ can be expressed as follows

$$\frac{\partial \zeta}{\partial \bar{p}_d} = \text{diag} \left\{ \frac{2}{u_{\max} - u_{\min}}, \frac{2}{v_{\max} - v_{\min}}, \dots, \frac{2}{v_{\max} - v_{\min}} \right\} \quad (68)$$

$$\frac{\partial \tilde{\varphi}}{\partial \zeta} = \frac{1}{(1 + \bar{\varphi})^2} f^T K \frac{\partial f}{\partial \zeta} \quad (69)$$

$$\frac{\partial f}{\partial \zeta} = \text{diag} \left\{ \frac{1 - \zeta_1^{2\kappa-1} \zeta_1^*}{(1 - \zeta_1^{2\kappa})^{(2\kappa+1)/2\kappa}}, \dots, \frac{1 - \zeta_8^{2\kappa-1} \zeta_8^*}{(1 - \zeta_8^{2\kappa})^{(2\kappa+1)/2\kappa}} \right\}. \quad (70)$$

It is clear from (62)-(70) that $\bar{p}_d(t) \rightarrow \bar{p}^*$ when $\nabla \varphi(\bar{p}_d) \rightarrow 0$.