Mechatronics 22 (2012) 410–422

Contents lists available at SciVerse ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Adaptive visual servo control to simultaneously stabilize image and pose error

N.R. Gans [a,*], G. Hu [b], J. Shen [a], Y. Zhang [a], W.E. Dixon [c]

[a] Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, United States
[b] EXQUISITUS, Centre for E-City, School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore
[c] Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, United States

ARTICLE INFO

Article history:
Available online 13 October 2011

Keywords:
Visual servoing
Nonlinear adaptive control

ABSTRACT

A novel, Lyapunov-based visual servo controller is presented that stabilizes both the entire image and pose error vectors simultaneously, rather than a subset of the errors. Furthermore, the controller uses adaptive depth estimation to eliminate the need to measure depth or obtain knowledge of the scene. A stability proof is presented. Simulation and experimental results compare the performance of the proposed method to PBVS, IBVS and 2.5D VS approaches.

1. Introduction

There are many methods of visual servo control that are classically grouped into image based visual servoing (IBVS) and position based visual servoing (PBVS) [1–3]. PBVS methods use image features to estimate the position and orientation (i.e., pose) error in the task space, and a camera velocity is generated from this error. IBVS methods measure an error in the image features, mapping desired feature point velocities to camera velocities.

IBVS and PBVS have well documented strengths and weaknesses [4]. The key strength of IBVS methods is the proclivity to maintain the feature points in the camera field of view. However, meeting the desired feature trajectories can require large camera movements that physical robots cannot meet. Furthermore, the control laws can suffer from unpredictable singularities in the image Jacobian. PBVS methods typically yield physically valid camera trajectories, with known (and hence avoidable) singularities and no local minima. However, there is no explicit control of the image features, and features may leave the field of view, resulting in task failure.

Previous attempts have been made to address these issues by combining IBVS and PBVS. There are several methods that partition the control along degrees of freedom, using IBVS for some velocity components and PBVS for the others [5–10]. A subset of these partitioned methods are referred to as 2.5D visual servoing [5,7], or as homography-based visual servoing [9,10], due to their reliance on the Euclidean homography [11] for pose estimation. These methods can generally provide asymptotic or exponential stability for a subset of image and pose errors. For some partitioned methods, it can be shown that if the subsets of image and pose error are zero, the entire error will be zero. However, not all Cartesian error or feature point error terms are explicitly controlled. During regulation the uncontrolled error terms may behave unpredictably.

There are also switched system controllers that actively switch between IBVS and PBVS depending on the current pose and/or state of the image features [12–14]. These methods typically can prevent failure, but cannot always guarantee asymptotic stability and may require a priori knowledge of the task to design the switching surfaces.

A third option is the use of navigation functions to generate desired feature trajectories which could account for constraints in the image and pose space [15,16]. These methods are asymptotically stable, but a priori knowledge of the scene/target and task space is necessary to guarantee a global minimum to the potential fields.

Other approaches have shown promise as well. A second order Hessian-like matrix is used in [17] rather than the typical first order Jacobian-like matrix. This approach reduces the pose problems associated with IBVS, but it requires ten feature points to create a full rank matrix. A spherical projection model is used in [18,19] to create IBVS controllers that do not experience severe coupling between translation and rotation.

A problem that both IBVS and PBVS systems face is the need for depth estimation and/or knowledge of the scene. IBVS methods require some amount of 3D knowledge of the depth to the target in every view. Some pose reconstruction methods [20] solve for depth to the target, but require detailed knowledge of the target to operate. Other pose reconstruction methods do not need a target model, [21,11], but require knowledge of the depth to the target in at least one view.

Visual servoing methods often assume depth is known when proving stability, and in experiments and implementation, the depth is estimated in a variety of ways. Recently, nonlinear adaptive control methods have been used to estimate or compensate for depth information in homography based visual servoing

* Corresponding author.
E-mail addresses: ngans@utdallas.edu (N.R. Gans), gqhu@ntu.edu.sg (G. Hu), shen.jinglin@utdallas.edu (J. Shen), yxz102220@utdallas.edu (Y. Zhang), wdixon@ufl.edu (W.E. Dixon).

0957-4158/$ - see front matter © 2011 Published by Elsevier Ltd.
doi:10.1016/j.mechatronics.2011.09.008

[22,10] and IBVS [23–26]. See [27] and references therein for a partial review of these methods.

In this paper, we present a controller to tackle the dual issues of stabilizing both the *entire pose error and entire image error* simultaneously, not just a subset of the pose and image errors. Furthermore, this controller requires no knowledge or measurement of depth to the targets. These control goals are accomplished through nonlinear control techniques that incorporate pose error, image error and adaptive depth estimation into a single controller. Lastly, no matrix inversion is necessary, so issues of matrix singularities are avoided. The pose and image errors are those used in standard IBVS and PBVS methods, and are calculated through well established methods. Integrator backstepping and Lyapunov-based stability analysis are used to develop the visual servo controller.

Hafez and Jawahar present a similar concept [28], utilizing image-based and position-based error vectors in a visual servo controller. The controller in [28] is a kinematic controller based on gradient descent and estimates feature depth through a particle filter-based approach. There is no stability analysis in [28], but simulations demonstrate its effectiveness. The controller presented in this paper is second order and developed using integrator backstepping and Lyapunov adaptive control techniques to account for unknown depth. A detailed stability analysis is presented for the method in this paper, as are experimental and simulated results. In addition to the novel controller presented, several properties of the IBVS and PBVS interaction matrices are presented that may be of value to researchers.

Section 2 presents background information and introduces notation. Section 3 briefly covers IBVS and PBVS, and introduces the adaptive depth estimation methods used in the controller. The proposed control law is introduced in Section 4, along with stability analysis. Finally, simulation and experiment results are provided to demonstrate the efficacy of the proposed method.

2. Model development

2.1. Robot motion model

Consider a camera mounted on the end-effector of a robot. A coordinate frame is rigidly attached to the camera at the focal point and oriented such that the z-axis is aligned with the optical axis. The pose of the camera frame with respect to the end-effector frame (i.e., the eye-to-hand calibration) is assumed to be constant and known. The Euclidean velocity of the camera is defined as

$$\xi(t) = [v(t)^T, \omega^T(t)]^T \in \mathbb{R}^6, \tag{1}$$

where $v(t) \in \mathbb{R}^3$ and $\omega(t) \in \mathbb{R}^3$ are the camera linear and angular velocity, respectively. The objective in eye-in-hand visual servoing is to find a velocity $\xi(t)$ such that the camera moves from its current pose $\mathcal{F}_c(t) \in SE(3)$ to a goal pose $\mathcal{F}_c^* \in SE(3)$. We assume that $\xi(t)$ is measurable.

The dynamics of the fully actuated robot system considered in this paper are given in generalized coordinates by

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau, \tag{2}$$

where $q(t) \in \mathbb{R}^n$ is the state in generalized coordinates (e.g., joint angles), $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ represents Coriolis and centripetal forces, $G(q) \in \mathbb{R}^n$ is the force of gravity, $F(\dot{q}) \in \mathbb{R}^n$ represents friction forces, and $\tau(t) \in \mathbb{R}^n$ denotes the generalized torques/forces. Using feedback linearization, the torque input[1]

$$\tau(t) = M\bar{u} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q})$$

yields

$$\ddot{q} = \bar{u}, \tag{3}$$

where $\bar{u}(t) \in \mathbb{R}^n$ is a subsequently designed control input. The joint angle velocity $\dot{q}(t) \in \mathbb{R}^n$ can be mapped to the Euclidean velocity $\xi(t)$ of a camera mounted to the robot end-effector as

$$\xi = J\dot{q}, \tag{4}$$

where $J(q) \in \mathbb{R}^{6\times n}$ is the manipulator Jacobian. The Jacobian is assumed to be continuously differentiable with a bounded derivative and the general inverse of $J(q)$ exists and is bounded, which is a standard assumption in visual servoing literature (e.g., see [30,31] and references therein).

2.2. Imaging model

The majority of visual servoing methods use points as image features. Consider a camera at a constant frame $\mathcal{F}_c^*$. The camera views a collection of k feature points in front of the camera. These points have coordinates

$$\bar{m}_j^* = \left[x_j^*, y_j^*, z_j^*\right]^T, \quad \forall j \in \{1,\ldots,k\} \tag{5}$$

in $\mathcal{F}_c^*$. An image of the points is captured, resulting in a projection to a set of points in the image plane. These image points are given by the normalized coordinates

$$m_j^* = \left[\frac{x_j^*}{z_j^*}, \frac{y_j^*}{z_j^*}, 1\right]^T, \quad \forall j \in \{1,\ldots,k\}. \tag{6}$$

Each normalized coordinate has a corresponding coordinate in image-space defined as

$$\left[u_j^*, v_j^*\right]^T = \left[\frac{x_j^*}{z_j^*}, \frac{y_j^*}{z_j^*}\right]^T, \quad \forall j \in \{1,\ldots,k\}.$$

Image points in digital images are typically expressed in pixel coordinates $p_j^* = \left[p_{xj}^*, p_{yj}^*, 1\right]^T \in \mathbb{R}^3$. Using standard projective geometry the relationship between p_j^* and m_j^* is given by

$$p_j^* = Am_j^*, \tag{7}$$

where A is a constant, invertible, upper-triangular camera calibration matrix [32]. Similarly, the Euclidean, normalized, and pixel coordinates of the feature points in a time-varying frame $\mathcal{F}_c(t)$ are respectively defined as

$$\bar{m}_j = [x_j, y_j, z_j]^T \quad m_j = \left[\frac{x_j}{z_j}, \frac{x_j}{z_j}, 1\right]^T$$

$$[u_j, v_j]^T = \left[\frac{x_j}{z_j}, \frac{y_j}{z_j}\right]^T \quad p_j = Am_j, \ \forall j \in \{1,\ldots,k\}.$$

Assume that $k \geqslant 4$ and all feature points reside in a plane π_s in front of the camera. These points have coordinates $\bar{m}_j^*, \ \forall j \in \{1,\ldots,k\}$ in the camera reference frame as in (5). The plane π_s has normal vector $-n^*$ in $\mathcal{F}_c^*$, and lies at a distance of $d^* > 0$ along n^* from the origin of $\mathcal{F}_c^*$. The camera is at a current pose $\mathcal{F}_c(t)$, separated from $\mathcal{F}_c^*$ by a rotation $R(t) \in SO(3)$ and translation $x(t) \in \mathbb{R}^3$.

A homography exists mapping m_j^* to $m_j(t)$, defined as the matrix $H(t) \in \mathbb{R}^{3\times 3}$ such that

$$m_j = \frac{z_j^*}{z_j} H m_j^* \tag{8}$$

$$m_j = \alpha_j \left(R + \frac{x}{d^*} n^{*T}\right) m_j^*. \tag{9}$$

[1] Feedback linearization of the robot dynamics is used for simplicity. A variety of techniques [29] could be applied to eliminate the assumption of an exact known model of the robot dynamics, or if the system is not feedback linearizable.

The matrix $H(t)$ can be decomposed to recover $R(t)$, $\frac{x(t)}{d^*}$, n^* and the ratio $\alpha_j(t) = \frac{z_j^*}{z_j(t)}$, $\forall j \in \{1, \ldots, k\}$ [11,33]. To simplify the notation, define the scaled translation $x_d(t) = \frac{x(t)}{d^*}$. Under the standard assumption that the feature points remain in front of the camera, $z_j(t)$ is bounded from below by a positive constant, and $\alpha_j(t)$ is bounded from above.

By using (7), the Euclidean relationship in (9) can be expressed in pixel coordinates as

$$p_j = \alpha_j AHA^{-1} p_j^* = \alpha_j G p_j^*. \tag{10}$$

Given knowledge of A and $k \geqslant 4$, it is possible to solve a set of linear equations for $G(t)$, and recover $H(t)$, $R(t)$, $\frac{x(t)}{d^*}$, n^* and $\alpha_j(t)$. Note that the translation error $x(t)$ can only be recovered up to the scale d^*, which is unknown without additional information.

3. Visual servoing

To provide context and motivation, and to highlight the contributions of the subsequent control development, a brief discussion of PBVS and IBVS techniques is presented. The basis for the adaptive depth compensation is also established.

3.1. Position based visual servoing

In PBVS, the camera and pose estimation algorithms act as a "Cartesian sensor", returning a pose estimate based on the current image and possibly geometric knowledge of the scene [2,3]. There are many methods of estimating the relative pose between $\mathcal{F}_c(t)$ and $\mathcal{F}_c^*$. For example, the methods presented in [34,11] require two images of the same scene taken at $\mathcal{F}_c(t)$ and $\mathcal{F}_c^*$, respectively, to recover the relative pose. The methods presented in [20,35] require an accurate target model to solve for relative pose between the camera and the target. The relative pose between the camera and the target can be used to recover the relative pose between $\mathcal{F}_c(t)$ and $\mathcal{F}_c^*$. The end result of any of these methods is the rotation $R(t)$ and translation $x(t)$ from $\mathcal{F}_c(t)$ to $\mathcal{F}_c^*$.

The pose error can be locally parameterized as a six element vector as

$$e_p = \left[x^T, \varphi u^T\right]^T, \tag{11}$$

where $\varphi(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}^3$ comprise the angle/axis representation of the rotation matrix $R(t)$. To avoid uniqueness problems associated with local rotation mappings, we adopt the convention that $\varphi(t) \in [-\pi, \pi)$ and $u(t)$ projects non-negatively onto the optical axis.

The time derivative of $e_p(t)$ is given as a function of the camera velocity $\xi(t)$ by

$$\dot{e}_p = L_p J \dot{q}; \tag{12}$$

where $L_p(t) \in \mathbb{R}^{6\times 6}$ is the Jacobian-like *pose interaction matrix* given by

$$L_p = \begin{bmatrix} R_{vc} & 0_{3\times 3} \\ 0_{3\times 3} & R_{vc} L_\omega \end{bmatrix}. \tag{13}$$

In (13), $R_{vc}(t) \in SO(3)$ is the rotation matrix from the frame in which $\xi(t)$ is measured to the camera frame. $R_{vc}(t)$ is identity if the camera frame and input velocity frame are the same. The matrix $L_\omega(t) \in \mathbb{R}^{3\times 3}$ maps angular velocity to $\frac{d}{dt}(u\varphi)$. As shown in [5], $L_\omega(t)$ is given by

$$L_\omega = I - \frac{\varphi}{2} u_\times + \left(1 - \frac{\operatorname{sinc}(\varphi)}{\operatorname{sinc}^2\left(\frac{\varphi}{2}\right)}\right) u_\times^2,$$

where I is the 3×3 identity matrix, $u_\times(t) \in \mathbb{R}^3$ is the skew symmetric matrix form of the vector $u(t)$, $\operatorname{sinc}(\varphi) = \frac{\sin(\varphi)}{\varphi}$, and $\operatorname{sinc}(0) = 1$.

When the Euclidean Homography in (9) is used to provide pose estimation, only the scaled translation vector $x_d(t)$ is obtained. The depth d^* must be known to recover $x(t) = d^* x_d(t)$. However, the depth is not generally known, so an estimate $\hat{d}^*(t)$ (which will be designed subsequently) is used. The estimated pose error can be expressed as

$$\hat{e}_p = \widehat{\Delta} e_{pd},$$

where $\widehat{\Delta}(t) = \begin{bmatrix} \hat{d}^*(t) I_3 & 0_3 \\ 0_3 & I_3 \end{bmatrix}$ and $e_{pd}(t) = [x_d(t)^T, u(t)^T \varphi(t)]^T$.

3.2. Image based visual servoing

With IBVS control, the control law is a function of an error that is measured in the image [1,3,36]. Given k feature points, the image-space error $e_i(t) \in \mathbb{R}^{2k}$ is defined

$$e_i(t) = [u_1(t), v_1(t), \ldots, u_k(t), v_k(t)]^T - \left[u_1^*, v_1^*, \ldots, u_k^*, v_k^*\right]^T. \tag{14}$$

The time derivative of $e_i(t)$ is given as a function of the camera velocity $\xi(t)$ by

$$\dot{e}_i = L_i \xi = L_i J \dot{q}, \tag{15}$$

where $L_i(t) \in \mathbb{R}^{2k\times 6}$ is the *image interaction matrix* (often referred to as the image Jacobian [1,3]) and is composed of k matrices $L_{ij}(t) \in \mathbb{R}^{2\times 6}$ corresponding to each feature point, concatenated above each other, where

$$L_{ij} = \begin{bmatrix} \frac{1}{z_j} & 0 & -\frac{u_j}{z_j} & -u_j v_j & 1 + u_j^2 & -v_j \\ 0 & \frac{1}{z_j} & -\frac{v_j}{z_j} & -1 - v_j^2 & u_j v_j & u_j \end{bmatrix}. \tag{16}$$

The matrix $L_i(t)$ depends on the feature point depths $z_j(t)$, which are generally unknown. If the feature points lie in a plane π_s in front of the camera, the homography $H(t)$ can be used to give the ratios $\alpha_j(t) = \frac{z_j^*}{z_j(t)}, j \in \{1, \ldots, k\}$, as in (8). Define a constant parameter $\theta_j = \frac{1}{z_j^*}$, then $\frac{1}{z_j(t)} = \alpha_j(t)\theta_j$ can be substituted into (16). Define the constant feature vector

$$\theta = [\theta_1, \ldots, \theta_k]^T = \left[\frac{1}{z_1^*}, \ldots, \frac{1}{z_k^*}\right]^T. \tag{17}$$

The *estimated image interaction matrix*, $\widehat{L}_i(t) \in \mathbb{R}^{2k\times 6}$, is composed of k matrices $\widehat{L}_{ij}(t) \in R^{2\times 6}$ defined by

$$\widehat{L}_{ij} = \begin{bmatrix} \alpha_j \hat{\theta}_j & 0 & -\alpha_j u_j \hat{\theta}_j & -u_j v_j & 1 + u_j^2 & -v_j \\ 0 & \alpha_j \hat{\theta}_j & -\alpha_j v_j \hat{\theta}_j & -1 - v_j^2 & u_j v_j & u_j \end{bmatrix},$$

where $\hat{\theta}_j(t) \in \mathbb{R} \forall j \in \{1, \ldots, k\}$ are subsequently designed adaptive estimates of the unknown constant parameters. The *submatrix estimation error* $\widetilde{L}_{ij}(t) \in \mathbb{R}^{2\times 6}$ is defined as

$$\widetilde{L}_{ij}(t) = L_{ij}(t) - \widehat{L}_{ij}(t).$$

The matrix $\widetilde{L}_{ij}(t)$ can be written as

$$\widetilde{L}_{ij} = \begin{bmatrix} \alpha_j \tilde{\theta}_j & 0 & -u_j \alpha_j \tilde{\theta}_j & 0 & 0 & 0 \\ 0 & \alpha_j \tilde{\theta}_j & -v_j \alpha_j \tilde{\theta}_j & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

$$= \widetilde{\Theta}_{M_j} L_{iMj}, \tag{19}$$

where $\tilde{\theta}_j(t) \in \mathbb{R}$, $\widetilde{\Theta}_{M_j}(t) \in \mathbb{R}^{2\times 2}$, and $L_{iMj}(t) \in \mathbb{R}^{2\times 6} \forall j \in \{1, \ldots, k\}$ are defined as

$$\tilde{\theta}_j = \theta_j - \hat{\theta}_j \qquad \widetilde{\Theta}_{M_j} = \begin{bmatrix} \tilde{\theta}_j & 0 \\ 0 & \tilde{\theta}_j \end{bmatrix} \tag{20}$$

$$L_{iMj} = \begin{bmatrix} \alpha_j & 0 & -u_j \alpha_j & 0 & 0 & 0 \\ 0 & \alpha_j & -v_j \alpha_j & 0 & 0 & 0 \end{bmatrix}. \tag{21}$$

Based on (20) and (21), the *matrix estimation error* $\widetilde{L}_i(t) \in \mathbb{R}^{2k\times 6}$ can be written as

$$\widetilde{L}_i = \widetilde{\Theta}_M L_{iM},$$

where $\widetilde{\Theta}_M(t) \in \mathbb{R}^{2k\times 2k}$ is a diagonal matrix composed of blocks $\widetilde{\Theta}_{M_j}(t)$, and $L_{iM}(t) \in \mathbb{R}^{2k\times 6}$ is composed of stacked $L_{iMj}(t)$ matrices. Finally, $e_i^T(t)\widetilde{\Theta}_M(t)\ L_{iM}(t)$ can be rewritten in terms of the adaptive estimation error vector $\tilde{\theta}(t) \in \mathbb{R}^k$ as

$$e_i^T \widetilde{\Theta}_M L_{iM} = \tilde{\theta}^T e_{iM} L_{iM},$$

where $e_{iM}(t) \in \mathbb{R}^{k\times 2k}, e_{uj}(t) \in \mathbb{R}$ and $e_{vj}(t) \in \mathbb{R} \forall j \in \{1 \dots k\}$ are given by

$$e_{iM} = \begin{bmatrix} e_{u1} & e_{v1} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & e_{u2} & e_{v2} & 0 & \cdots & \cdots & 0 \\ & & & \ddots & \ddots & & & \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & e_{uk} & e_{vk} \end{bmatrix}$$

$$e_{uj} = u_j - u_j^* \qquad e_{vj} = v_j - v_j^*.$$

4. Simultaneous image and position visual servoing

Efforts in this paper seek to design a controller which can asymptotically reduce the image error and pose error simultaneously. In this way, the system will not encounter unbounded image space errors or unbounded camera motions. Furthermore, we seek a controller that is stable with no knowledge of the 3D depths of the feature points. We refer to this control method as *Simultaneous Image/Pose Visual Servoing* (SIPVS).

4.1. Controller development

After using (4), (15) and (12), the time derivatives of the image and pose errors are given as functions of the joint velocities as

$$\dot{e}_i = L_i \xi = L_i J \dot{q} \text{ and } \dot{e}_p = L_p \xi = L_p J \dot{q}.$$

Motivated by the desire to include the robot dynamics, a desired camera velocity $\xi_d(t) \in \mathbb{R}^6$ is injected into the open-loop image and pose error system as

$$\dot{e}_i = L_i \tilde{\xi} + L_i \xi_d \text{ and } \dot{e}_p = L_p \tilde{\xi} + L_p \xi_d, \tag{22}$$

where $\tilde{\xi}(t) \in \mathbb{R}^6$ is the backstepping error defined as

$$\tilde{\xi} = J\dot{q} - J\dot{q}_d = J\dot{q} - \xi_d. \tag{23}$$

Based on the open-loop error systems in (22), the desired velocity is designed as

$$\xi_d = -k\widehat{L}_i^T e_i - kL_p^T \widehat{\Delta} e_{pd}, \tag{24}$$

where k is a positive scalar control gain. Substituting (24) into (22) gives the closed-loop error system

$$\dot{e}_i = -kL_i\widehat{L}_i^T e_i - kL_i L_p^T \widehat{\Delta} e_{pd} + L_i \tilde{\xi} \tag{25}$$

$$\dot{e}_p = -kL_p L_p^T \widehat{\Delta} e_{pd} - kL_p \widehat{L}_i^T e_i + L_p \tilde{\xi}. \tag{26}$$

The open-loop system for the backstepping error $\tilde{\xi}(t)$ can be developed as

$$\dot{\tilde{\xi}} = \dot{J}\dot{q} + J\ddot{q} - \dot{\xi}_d(e_i, e_p) = \dot{J}\dot{q} + J\bar{u} - \dot{\xi}_d(e_i, e_p). \tag{27}$$

Based on (27), the controller $\bar{u}(t)$ is designed as

$$\bar{u} = J^+\left(-\dot{J}\dot{q} + \dot{\xi}_d - (k_\xi + k_v)\tilde{\xi} - 2k_v\xi_d - \widehat{L}_i^T e_i - L_p^T \widehat{\Delta} e_{pd}\right), \tag{28}$$

where $J^+(q)$ is the general inverse of $J(q)$ such that $J(q)J^+(q) = I$, and $k_\xi,\ k_v \in \mathbb{R}$ are positive, constant, scalar control gains. In (28), $\dot{\xi}_d(t)$ can be expanded as

$$\begin{aligned} \dot{\xi}_d &= -k\dot{\widehat{L}}_i^T \dot{e}_i - k\dot{\widehat{L}}_i^T e_i - kL_p^T(\dot{\widehat{\Delta}} e_{pd} + \widehat{\Delta}\dot{e}_{pd}) - k\dot{L}_p^T \widehat{\Delta} e_{pd} \\ &= k\dot{\widehat{L}}_i^T \left(kL_i\widehat{L}_i^T e_i + kL_i L_p^T \widehat{\Delta} e_{pd} - L_i\tilde{\xi}\right) - k\dot{\widehat{L}}_i^T e_i - k\dot{L}_p^T \widehat{\Delta} e_{pd} \\ &\quad - kL_p^T \dot{\widehat{\Delta}} e_{pd} - kL_p^T \widehat{\Delta}\dot{e}_{pd}. \end{aligned} \tag{29}$$

The term $\dot{e}_{pd}(t)$ is measured through discrete methods such as backwards differencing. Numerical differentiation will inject measurement noise, and the development of an output feedback controller is an open topic for future work. The matrices $\dot{\widehat{L}}_i^T(t)$ and $\dot{\widehat{\Delta}}(t)$ are functions of the adaptive estimate signals $\dot{\hat{\theta}}(t)$ and $\dot{\hat{d}}^*(t)$ and bounded, measurable signals, as shown in Appendix B.

The first row of (28) is based on feedback linearization methods to remove unwanted nonlinear terms. The second row of (28) is used to cancel out cross terms in the subsequent stability analysis. Based on the subsequent stability analysis, the adaptive estimation update laws are designed as

$$\dot{\hat{\theta}} = \Gamma e_{iM} L_{iM}\left(\tilde{\xi} - k\widehat{L}_i^T e_i - kL_p^T \widehat{\Delta} e_{pd}\right) \tag{30}$$

$$\dot{\hat{d}}^* = \gamma\left(\tilde{\xi}_3^T - k\hat{d}^* x_d^T - k\left(e_i^T \widehat{L}_i\right)_3 R_{vc}\right) x_d, \tag{31}$$

where $\tilde{\xi}_3(t) \in \mathbb{R}^3$ and $(e_i^T\widehat{L}_i)_3(t) \in \mathbb{R}^{1\times 3}$ are the first three elements of the vectors $\tilde{\xi}(t)$ and $e_i^T(t)\widehat{L}_i(t)$, respectively, and $\gamma \in \mathbb{R}$ and $\Gamma \in \mathbb{R}^{3\times 3}$ are constant gains.

After substituting (28) into (27), the closed-loop error system for $\dot{\tilde{\xi}}(t)$ can be obtained as

$$\begin{aligned} \dot{\tilde{\xi}} &= -k_\xi\tilde{\xi} - k_v\tilde{\xi} - 2k_v\xi_d - \widehat{L}_i^T e_i - L_p^T \widehat{\Delta} e_{pd} \\ &= -(k_\xi + k_v)\tilde{\xi} + (2k_v k - 1)\widehat{L}_i^T e_i + (2k_v k - 1)L_p^T \widehat{\Delta} e_{pd}. \end{aligned} \tag{32}$$

4.2. Closed-loop analysis

To demonstrate the stability properties of the closed loop system, two standard assumptions are made.

Assumption 1. $L_i(t)$ is full rank at the goal pose.

Assumption 2. The control gains satisfy the conditions

$$k_v < \frac{1}{k}. \tag{33}$$

Remark 1. If $L_p(t)$ and $\widehat{\Delta}(t)$ are both full rank for all t, then $\widehat{\Delta}(t)L_p(t)e_p(t) = 0$ if and only if $e_p(t) = 0$.

Lemma 1. *If $L_i(t)$ is full rank, and no element of $\hat{\theta}(t) = 0$, then $\widehat{L}_i(t)$ is full rank.*

Proof. The proof is given in Appendix C. □

Remark 2. If $\widehat{L}_i(t)$ is full rank for all t, then $\widehat{L}_i^T(t)e_i(t) = 0$ if and only if $e_i(t) = 0$.

Theorem 2. *Under Assumptions 1 and 2, the controller in (28), along with the depth estimation laws in (30) and (31), stabilizes the system such that $e_p(t)$ and $e_i(t)$ are bounded and either $e_p(t) \to 0$ and $e_i(t) \to 0$ or the system converges to an equilibrium point such that $L_p^T(t)\widehat{\Delta}(t)e_{pd}(t) + \widehat{L}_i^T(t)e_i(t) \to 0$ as $t \to \infty$.*

Proof. Consider the positive definite Lyapunov function

$$V = \frac{1}{2}e_i^T e_i + \frac{1}{2}e_p^T e_p + \frac{1}{2}\tilde{\xi}^T\tilde{\xi} + \frac{1}{2}\tilde{\theta}^T\Gamma^{-1}\tilde{\theta} + \frac{1}{\gamma}\tilde{d}^{*2} \tag{34}$$

with derivative

$$\dot{V} = e_i^T\dot{e}_i + e_{pd}^T\Delta\dot{e}_p + \tilde{\xi}\dot{\tilde{\xi}} - \tilde{\theta}^T\Gamma^{-1}\dot{\hat{\theta}} - \frac{1}{\gamma}\tilde{d}^*\dot{\tilde{d}}^*. \tag{35}$$

As shown in Appendix A, $\dot{V}(t)$ in (35) can be written as

$$\dot{V} = -(k - k^2k_v)\left(L_p^T\widehat{\Delta}e_{pd} + \widehat{L}_i^T e_i\right)^T\left(L_p^T\widehat{\Delta}e_{pd} + \widehat{L}_i^T e_i\right) - k_\xi\tilde{\xi}^T\tilde{\xi} - k_v\xi^T\xi. \tag{36}$$

Based on (36) and the gain condition in (33), $\dot{V}(t) \leqslant 0$. From these results, and the boundedness properties given in Appendix B, the Corollary to Barbalat's Lemma [37] can be used to conclude that $\tilde{\xi}(t) \to 0$ and $L_p^T(t)\widehat{\Delta}(t)e_{pd}(t) + \widehat{L}_i^T(t)e_i(t) \to 0$ as $t \to \infty$.

Within the domain $\theta \in [-\pi, \pi)$, $L_p(t)$ is full rank. By using the well-known projection operator in the adaptive law (31), we can ensure that $\hat{d}^*(t) > 0$ for all t.This implies that $\widehat{\Delta}(t)$ is full rank for all t. From Remark 1, then $\widehat{\Delta}(t)L_p(t)e_p(t) = 0$ if and only if $e_p(t) = 0$.

From Assumption 1, $L_i(t)$ is full rank at the origin implies $L_i(t)$ full rank in some neighborhood of the origin. The projection operation can be used on $\hat{\theta}(t)$ to ensure that its elements are nonzero. Therefore, from Lemma 1 and Remark 2, there exists neighborhood of the origin where $\widehat{L}_i^T(t)e_i(t) = 0$ if and only if $e_i(t) = 0$.

The fact that $L_p^T(t)\widehat{\Delta}(t)e_{pd}(t) + \widehat{L}_i^T(t)e_i(t) \to 0$ and $\tilde{\xi}(t) \to 0$ implies that $\xi_d(t) \to 0$ and $\xi(t) \to 0$, so the system converges to an equilibrium point. There are three possibilities concerning this equilibrium point:

- **Case 1**: An equilibrium point exists at the goal (i.e., $e_i(t) \to 0$ and $e_p(t) \to 0$).
- **Case 2**: An equilibrium point can exist if $e_{pd}(t) = -\widehat{\Delta}^{-1}(t)L_p^{-T}(t)\widehat{L}_i^T(t)e_i(t) \neq 0$.
- **Case 3**: An equilibrium point can exist if $e_i(t)$ is in the nullspace of $\widehat{L}_i^T(t)$, and $e_p(t)$ is in the nullspace of $L_p^T(t)\widehat{\Delta}(t)$ at the same time. That is, $L_p^T(t)\widehat{\Delta}(t)e_{pd} = 0$ and $\widehat{L}_i^T(t)e_i(t) = 0$.

However, it was shown that $L_p^T(t)\widehat{\Delta}(t)$ is full rank in the domain $\theta \in [-\pi, \pi)$. If $e_i(t)$ is in the nullspace of $\widehat{L}_i^T(t)$, then $L_p^T(t)\widehat{\Delta}(t)e_{pd} + \widehat{L}_i^T(t)e_i(t) \to 0$ implies $e_p(t) \to 0$. Therefore, there is no equilibrium point where $L_p^T(t)\widehat{\Delta}(t)e_{pd} = 0$ and $\widehat{L}_i^T(t)e_i(t) = 0$ except the goal position. Thus, case 3 is impossible. This leaves the two possible equilibrium points described in the theorem. □

4.3. Discussion of stability analysis

Let $\mathcal{D}_i \subset SE(3)$ be the region where IBVS is asymptotically stable for $e_i(t)$. IBVS can also be shown to be asymptotically stable for both $e_i(t)$ and $e_p(t)$ simultaneously in a neighborhood of $e_p(t) = 0$ denoted $\mathcal{D}_{ip} \subset \mathcal{D}_i$. This neighborhood includes the space of pure translations and translations with suitably small rotations [12]. Let $\mathcal{D}_p \subset SE(3)$ be the region where PBVS is asymptotically stable for $e_p(t)$. In a neighborhood of $e_p(t) = 0$, denoted $\mathcal{D}_{pi} \subset \mathcal{D}_p$, PBVS can be shown to be asymptotically stable for $e_i(t)$ and $e_p(t)$ simultaneously. This neighborhood includes the space of pure translations and translations with suitably small rotations. Define $\mathcal{D}^* = \mathcal{D}_{ip} \cap \mathcal{D}_{pi}$. $\mathcal{D}^*$ is nonempty, and includes the space of pure translations and translations with suitably small rotations. The relationships between stability regions is illustrated in Fig. 1.

For SIPVS, we have proven that the pose error and image error simultaneously converge to zero, or the system converges to some equilibrium point where $L_p^T(t)\widehat{\Delta}(t)e_{pd} + \widehat{L}_i^T(t)e_i(t) = 0$. The existence of such an equilibrium cannot, at this time, be proven or disproven, but no such equilibrium point has been encountered in simulations and experiments. The future efforts will focus on investigating the existence of the equilibrium point, particularly in the set $\mathcal{D}^*$.

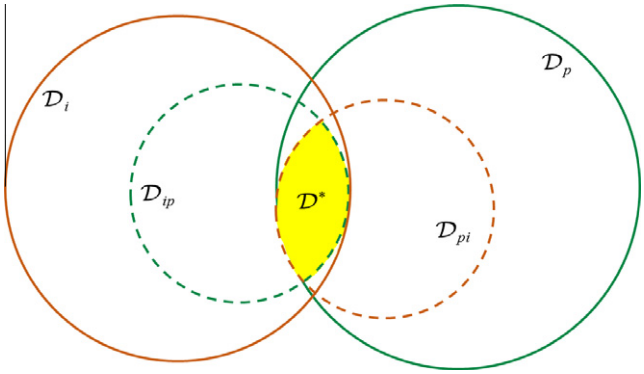


Fig. 1. Relation between stability regions.

The SIPVS method has several strengths compared to IBVS, PBVS, and methods such as 2.5D VS. The entire pose error vector and entire image error vector for all points are included in the control law. This means that all terms are stabilized, not just a subset. The adaptive depth estimation enables the system to regulate the error with no depth measurements or knowledge of the object. The SIPVS approach also does not require a matrix inverse, reducing computational complexity and eliminating singularities issues that can plague IBVS approaches.

Errors in the calibration matrix A are known to affect the accuracy of the estimation of H and affect stability of the control system [5]. Research has been performed to address an unknown calibration matrix, including adaptive control approaches [10]. The SIPVS could likely be extended to include these additional adaptive control parameters. Measurement noise is also known to affect accuracy of VS, particularly methods that use the homography. The experimental results of Section 6 show that the method works well, even when faced with noise in feature point tracking. The computational complexity of SIPVS is of the same order as other methods that use the homography matrix (e.g., PBVS and 2.5D). Since no matrix inversion is needed, computational complexity is lighter than IBVS for large number of points.

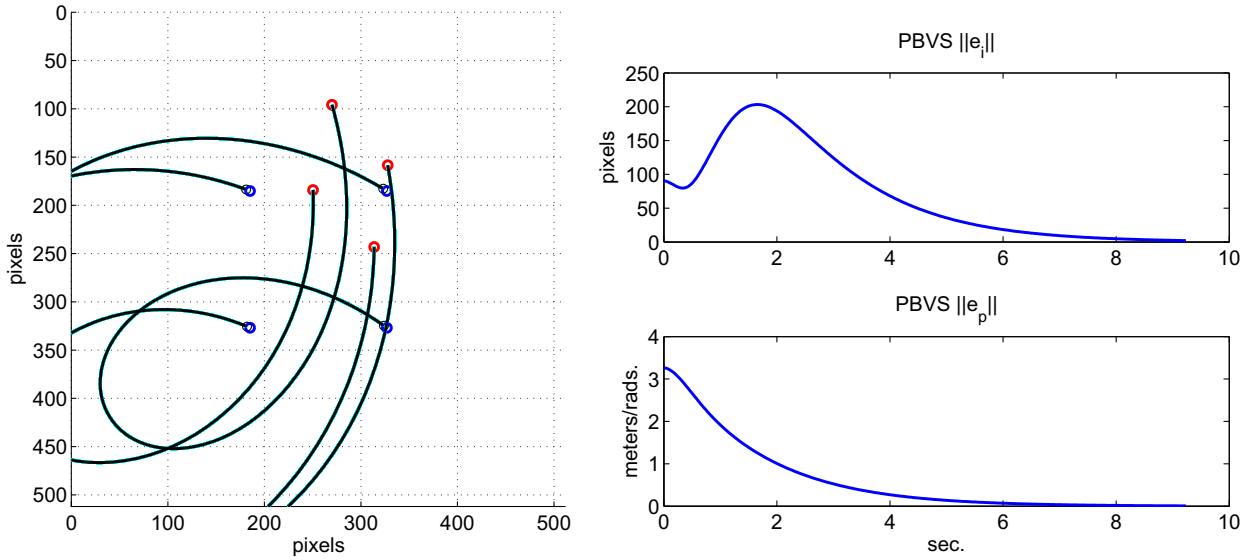
5. Simulation results

This section presents simulation results for the developed SIPVS system. Typical IBVS and PBVS systems are simulated for the same task for comparison, with PBVS using the Euclidean Homography. 2.5D VS is included as well, for comparison with a well-regarded mixed VS approach [5].

IBVS, PBVS and 2.5D methods are extended to a second order regulation problem, since SIPVS is a second order system. Furthermore, perfect knowledge of depth is given to the IBVS, PBVS and 2.5D systems, while SIPVS uses adaptive depth compensation. Gains for all systems were chosen such that the presented task finished in approximately 9.5 s. The task presented is an initial pose error of

$$[x^T u^T \varphi] = [0.12, -0.27, -0.16, 1.16, -0.54, -2.44]$$

in meters and radians, and a desired pose error of a zero vector. Large rotations are known to be problematic for both IBVS and PBVS, so this is a difficult task.



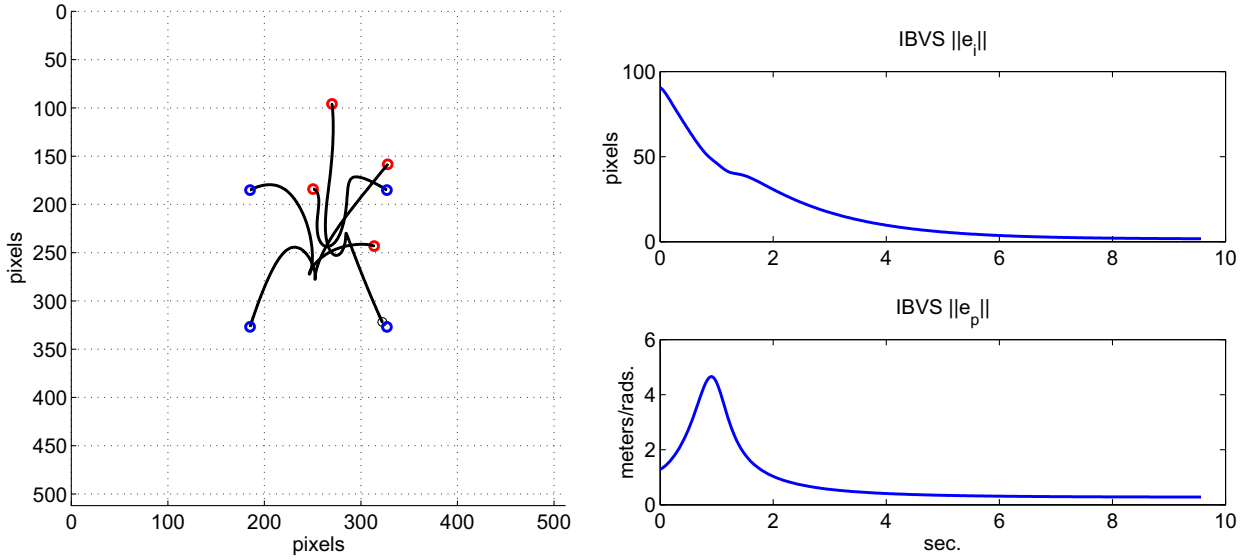
(a)Feature point trajectories (b)Image and pose error norms

Fig. 2. Simulation results for PBVS – The features points leave the field of view, which would likely cause failure in implementation. The corresponding increase in image error can be seen, though the pose error decrease monotonically.

The feature point trajectory and error norms over time for PBVS are given in Fig. 2a and b, respectively. The same data for IBVS are given in Fig. 3a and b. PBVS shows a large increase in the image error, such that the features leave the field of view. IBVS shows a large increase in the pose error, such that the system could encounter task space limits or joint limits. The results for 2.5D VS are given in Fig. 4. 2.5D performs well, with strictly decreasing errors in both $\|e_i\|$ and $\|e_p\|$. We note that in this result, neither IBVS nor 2.5D VS show the characteristic straight line trajectories for one or more feature points, due to the large initial error and possibly the extension to second order systems. The controlled feature point in 2.5D VS is the one starting closest to the top of the image in Fig. 4a, which does follow a fairly straight path. All systems perform as expected in terms of image and pose error.

The trajectory of the feature points under SIPVS is shown in Fig. 5a, and the norms of the errors over time are given in Fig. 5b. The pose error monotonically decreases, and image error decreases but not monotonically.

Clearly, the SIPVS outperforms IBVS and PBVS in this difficult task, and compares favorably with 2.5D VS. This is especially strong when considering that the first three systems had perfect depth knowledge, while SIPVS used the proposed depth estimation. While overall convergence time is the same for all systems, the 2.5D VS rate of decrease is somewhat less steep than for SIPVS. For this particular task, 2.5D also brings a feature point closer to the edge of the image. It should be noted that this is not the controlled feature point, which follows a short path to its goal.



(a)Feature point trajectories (b)Image and pose error norms

Fig. 3. Simulation results for IBVS – All features remain in view, but the camera must retreat four meters to achieve this. The increase in pose error is clearly seen.

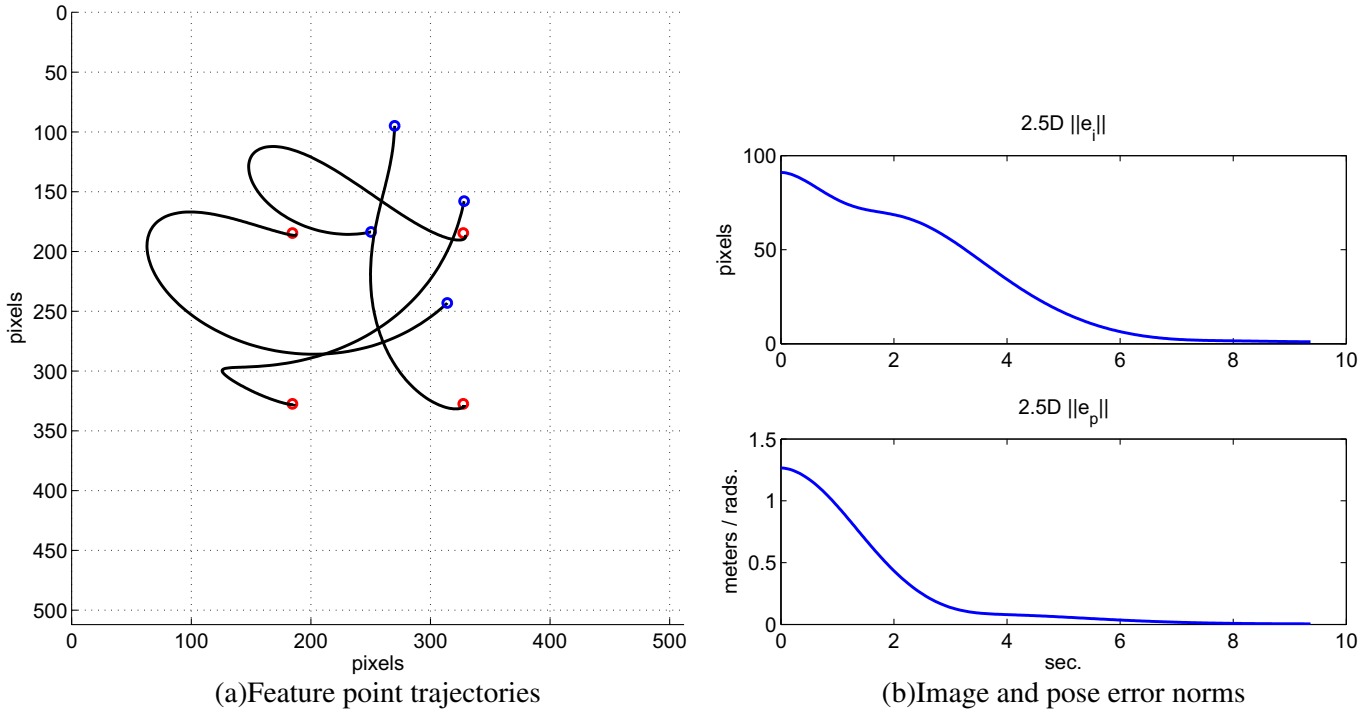


Fig. 4. Simulation results for 2.5D VS – All features remain in view, and there is no apparent camera retreat.

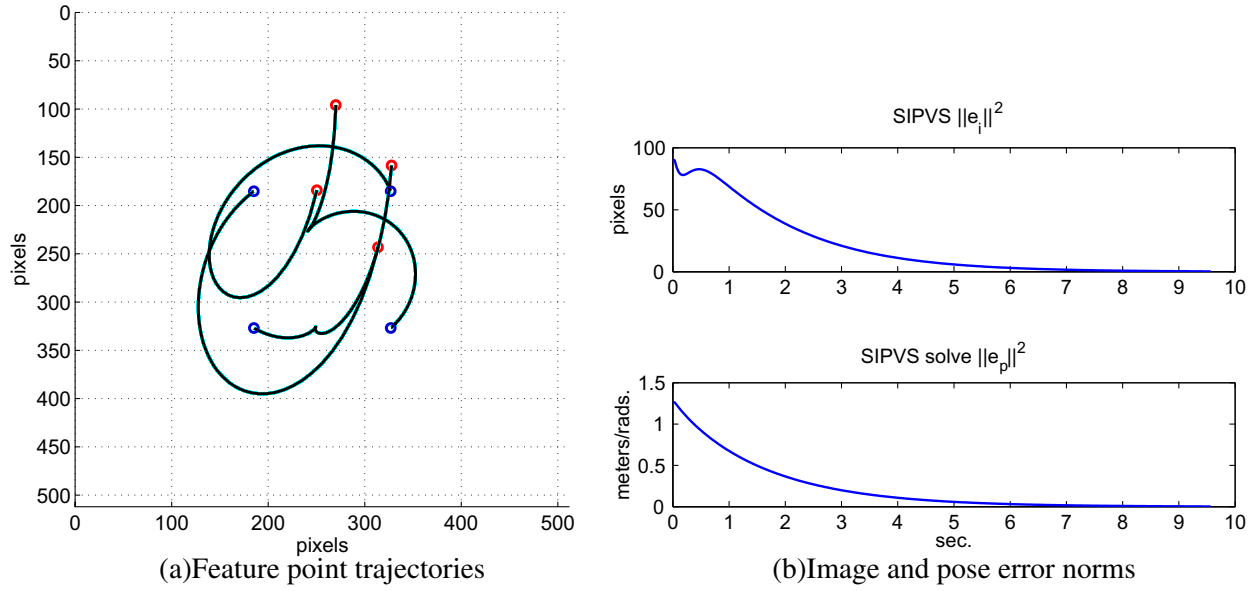
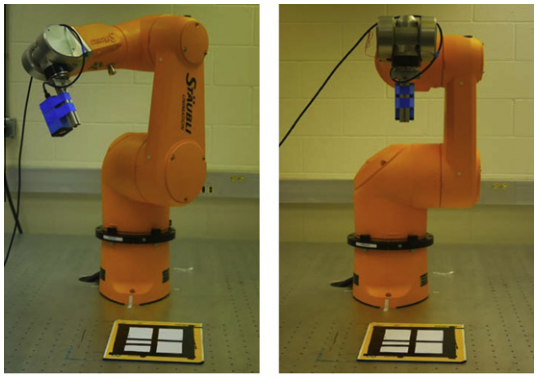


Fig. 5. Simulation results for SIPVS – Feature points remain in view, and both image and pose error decrease to zero. There is a small increase in image error, likely due to the depth estimate not yet having converged.

6. Experiment results

This section presents experiment results of the proposed SIPVS system. The Experiment setup is shown in Fig. 6. A six degree of freedom Staubli TX90 robot arm and a calibrated camera with fixed focal length are used. The extrinsic (eye-to-hand) calibration of the camera is done using the method of Tsai et al. [38]. The same task is performed with IBVS, PBVS and SIPVS. The initial pose error (shown in Fig. 6a) is $[x^T, u^T\varphi] = [-250, -200, -100, 0.5749, 0.0859, 1.0954]$ in millimeters and radians, measured in the goal camera frame. The goal pose is shown in Fig. 6b). The Lucas–Kanade method is used to track feature points on a planar target. PBVS and SIPVS use the Euclidean Homography matrix. The constant depth at the goal d^* is given as known a condition for PBVS. Adaptive estimation is performed for IBVS and SIPVS using the estimates detailed in this paper.

Figs. 7 and 8 give the experiment results using PBVS and IBVS. The feature point trajectory, pose error and image error norms for PBVS are given in Figs. 7a–c, respectively. It can be seen that pose error decreases rapidly at the beginning. However, instead of going to zero, the position error converge to about 20 mm after a small increase. On the other hand, the image error increases by almost 100 pixels at the beginning. The feature point trajectories also show a large curve, which agrees with the simulation results for PBVS. The relatively large residue error for PBVS is probably



(a)Initial pose (b)Goal pose

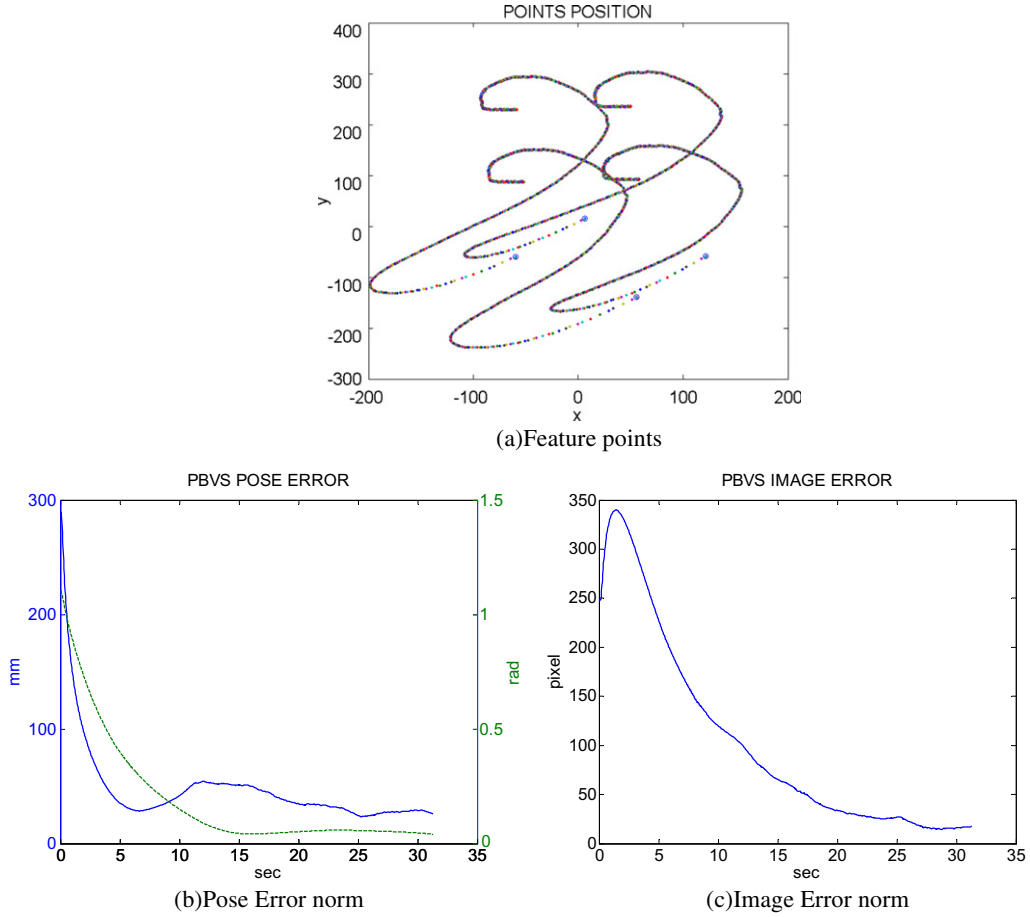
Fig. 6. The experiment setup – The camera is mounted to the robot wrist and looks at a planar target in the initial and goal configurations.

due to the fact that PBVS is sensitive to both intrinsic and extrinsic calibration errors. Also, since motion reconstruction using the homography matrix can be effected by image noise, the error trajectories for PBVS include the effects of noise.

Image trajectory and error norms for IBVS are given in Fig. 8. Note that the IBVS experiment stops after about 12 s because the robot arm hits its joint limit due to camera retreat. Pose error increases after 4 s until hitting the joint limit. Unlike in the simulation, where the image error decreases monotonically, the image error increases initially. This is likely due to the fact that the adaptive depth estimates had not yet converged. Fig. 10a shows the depth estimates for IBVS over time. As expected, the depth estimates converge to constant values such that the system is stable. Note that the adaptive estimation methods can stabilize the system, though the estimate is not guaranteed to converge to the true depth in the absence of a persistently exciting camera motion.

Experimental results of proposed SIPVS method are presented in Figs. 9 and 10. Image trajectory and error norms for SIPVS are given in Fig. 9. Depth estimation is shown in Fig. 10b. It can be seen that the pose error decreases monotonically, and image error decreases quickly but experiences a small increase between 8 and 12 s of operation. A possible reason is that the chosen gains cause the image error converge faster than the pose error. The larger pose error may dominate the velocity at around 8 s, compelling the camera to the direction that increases the image error. The final error residue is much smaller than PBVS and IBVS. Overall, the experiments for all three systems agree with our expectations and the simulation results. SIPVS shows better performance than PBVS and IBVS, and the adaptive depth estimate works well.

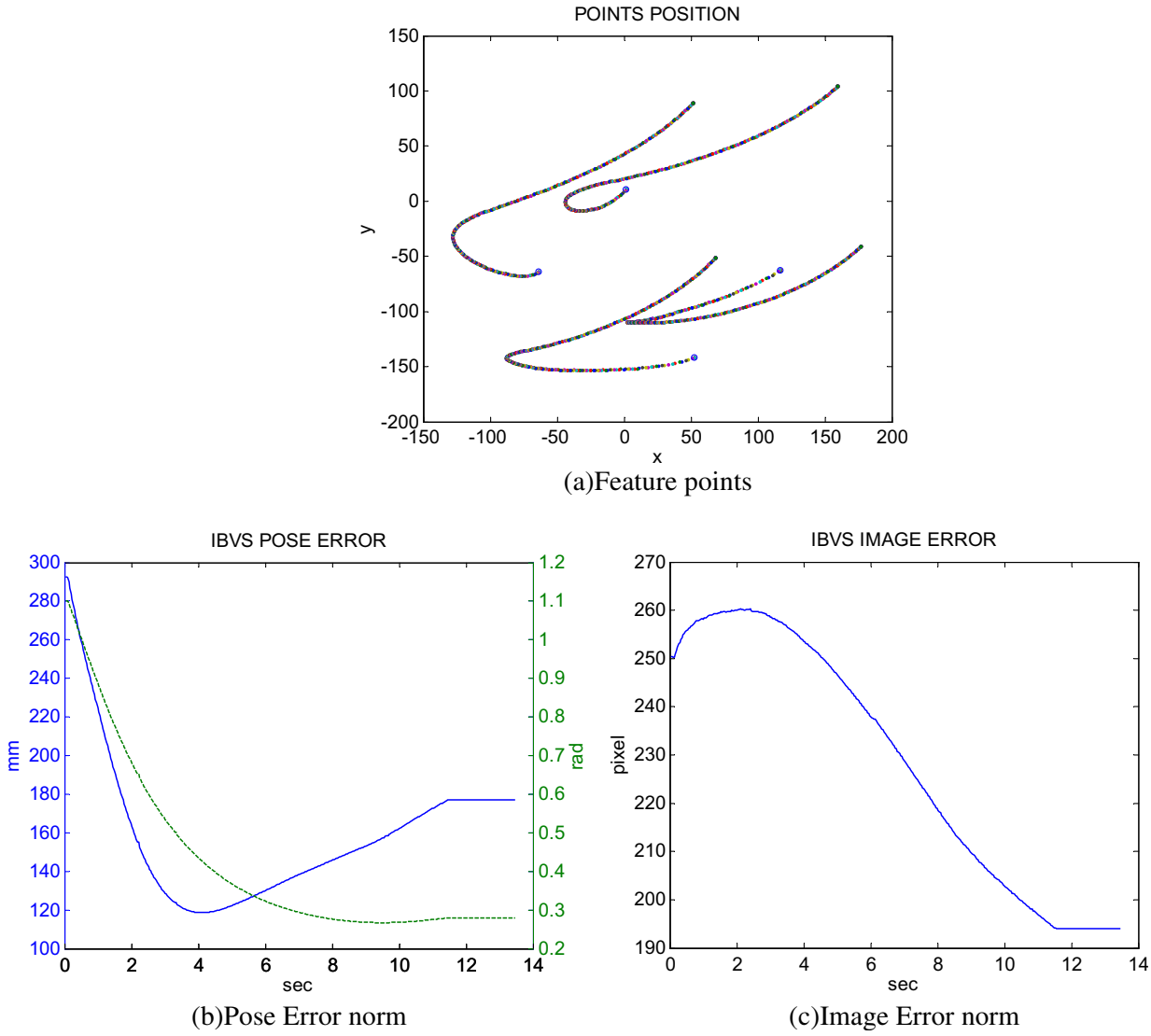
Similar to other visual servo methods that use the Euclidean Homography, SIPVS is sensitive to noise. If the image noise is high, the final pose and image error may not converge to zero, since pose information cannot be perfectly recovered. Nevertheless, the proposed system appears to work well in experiments that include feature noise. Methods have been proposed in recent years to estimate the homography matrix that are robust to noise, such as the direct visual servoing in [39].



(a)Feature points

(b)Pose Error norm (c)Image Error norm

Fig. 7. Experiment results for PBVS – The pose error decreases nicely, but the image error initially increases by almost 100 pixels. This is seen in large feature point motions in the image.



(a)Feature points

(b)Pose Error norm

(c)Image Error norm

Fig. 8. Experiment results for IBVS – The image error undergoes an overall decrease, though there is an initial increase, possibly due to the adaptive depth estimates taking some time to converge. The position error increases until the robot reaches its joint limits and the experiment fails.

7. Conclusions and future work

We have presented a novel visual servo controller that incorporates nonlinear control techniques to regulate both the pose error and image error simultaneously while estimating unknown depth parameters. This work was inspired by the well known weaknesses of IBVS and PBVS methods, which have fueled much previous work. The contribution here is that the entire image error and pose error are simultaneously stabilized, rather than partitioning the controller such that only parts of the image error or pose error are explicitly regulated. Furthermore, this controller uses adaptive depth estimation such that no measurement of the depth or knowledge of the scene is needed. There is also no matrix inversion necessary for the vision-based control.

There are several avenues for future work. The system can only be proven stable at this point, as there may be equilibrium points other than the origin. If these equilibrium points can be proven to not exist, or proven to be unstable, then the system must converge to the origin. Additional attention can also be given to the specific IBVS and PBVS methods utilized. Different image features, image error measurements, pose reconstruction techniques, and representations of the pose errors could all give different results.

Appendix A. Closed-loop stability analysis details

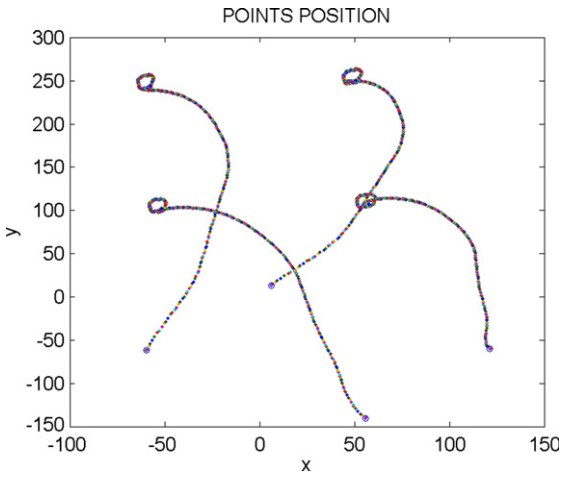
Substituting (26), (25) and (32) into (35) gives

$$\begin{aligned}\dot{V} = &-ke_i^T\widehat{L}_i\widehat{L}_i^Te_i - ke_{pd}^T\Delta L_p\widehat{L}_p^T\widehat{\Delta}e_{pd} - (k_\xi + k_v)\tilde{\xi}^T\tilde{\xi}\\ &- ke_{pd}^T\Delta L_p\widehat{L}_i^Te_i - ke_{pd}^T\widehat{\Delta}L_pL_i^Te_i + \tilde{\xi}^TL_i^Te_i + (2k_vk-1)\tilde{\xi}^T\widehat{L}_i^Te_i\\ &+ \tilde{\xi}^TL_p^T\Delta e_{pd} + (2k_vk-1)\tilde{\xi}^TL_p^T\widehat{\Delta}e_{pd} - \tilde{\theta}^T\Gamma^{-1}\dot{\hat{\theta}} - \frac{1}{\gamma}\tilde{d}^*\dot{\hat{d}}^*. \end{aligned} \tag{A.1}$$

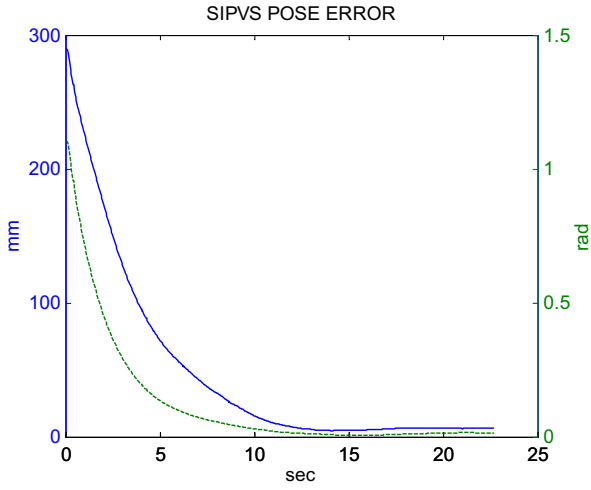
After substituting $L_i(t) = \widehat{L}_i(t) + \widetilde{L}_i(t)$, $\Delta(t) = \widehat{\Delta}(t) + \widetilde{\Delta}(t)$, grouping quadratic terms, and canceling common terms, (A.1) can be rewritten as

$$\begin{aligned}\dot{V} = &-ke_i^T\widehat{L}_i\widehat{L}_i^Te_i - ke_{pd}^T\widehat{\Delta}L_p\widehat{L}_p^T\widehat{\Delta}e_{pd} - (k_\xi + k_v)\tilde{\xi}^T\tilde{\xi} - \tilde{\theta}^T\Gamma^{-1}\dot{\hat{\theta}}\\ &- \frac{1}{\gamma}\tilde{d}^*\dot{\hat{d}}^* - ke_i^T\widetilde{L}_i\widehat{L}_i^Te_i - ke_{pd}^T\widetilde{\Delta}L_p\widehat{L}_p^T\widehat{\Delta}e_{pd} - ke_{pd}^T\widetilde{\Delta}L_p\widehat{L}_i^Te_i\\ &- ke_{pd}^T\widehat{\Delta}L_p\widetilde{L}_i^Te_i + \tilde{\xi}^T\widetilde{L}_i^Te_i + 2k_vk\tilde{\xi}^T\widehat{L}_i^Te_i - 2ke_{pd}^T\widehat{\Delta}L_p\widehat{L}_i^Te_i\\ &+ \tilde{\xi}^TL_p^T\widetilde{\Delta}e_{pd} + 2k_vk\tilde{\xi}^TL_p^T\widehat{\Delta}e_{pd}. \end{aligned} \tag{A.2}$$

By adding and subtracting $k_vk^2e_i^T\widehat{L}_i\widehat{L}_i^Te_i, k_vk^2e_{pd}^T\widehat{\Delta}L_pL_p^T\widehat{\Delta}e_{pd}$ and $2k_vkk_pe_{pd}^T\widehat{\Delta}L_p\widehat{L}_i^Te_i$, (A.2) can be expressed as



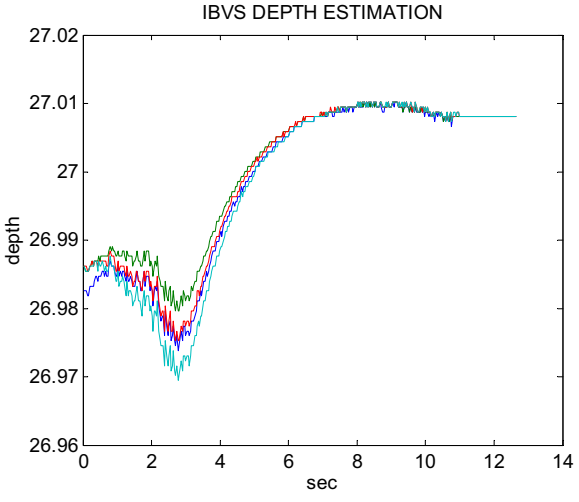
(a)Feature points



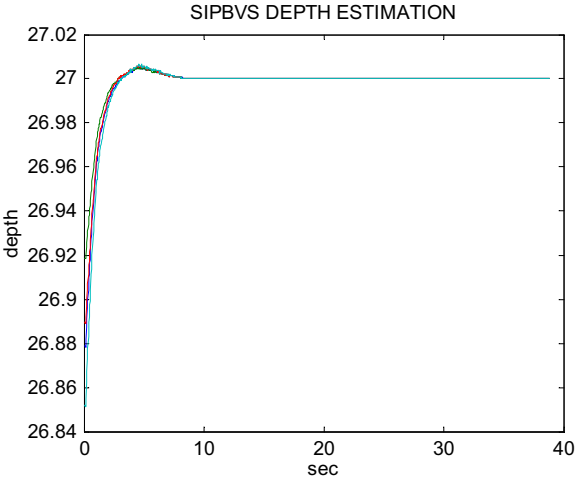
(b)Pose Error norm

(c)Image Error norm

Fig. 9. Experiment results for SIPVS – Both the image and pose error decreases quickly. There is much less motion in the feature point trajectories than PBVS.



(a)IBVS



(a)SIPVS

Fig. 10. Depth estimates for IBVS and SIPVS. Both systems converge smoothly to an estimate of about 27 mm.

$$\dot{V} = -(k-k_v k^2)e_i^T\widehat{L}_i\widehat{L}_i^T e_i - k_\xi\tilde{\xi}^T\tilde{\xi} - (k-k_v k^2)e_{pd}^T\widehat{\Delta}L_pL_p^T\widehat{\Delta}e_{pd} - \tilde{\theta}^T\Gamma^{-1}\dot{\hat{\theta}} - \frac{1}{\gamma}\tilde{d}^*\dot{\hat{d}}^* - ke_i^T\widetilde{L}_i\widehat{L}_i^T e_i - ke_{pd}^T\widetilde{\Delta}L_pL_p^T\widehat{\Delta}e_{pd} - ke_{pd}^T\widehat{\Delta}L_p\widehat{L}_i^T e_i - ke_{pd}^T\widehat{\Delta}L_p\widetilde{L}_i^T e_i + \tilde{\xi}^TL_p^T\widetilde{\Delta}e_{pd} + \tilde{\xi}^T\widetilde{L}_i^T e_i + 2k(k_vk-1)e_{pd}^T\widehat{\Delta}L_p\widehat{L}_i^T e_i - k_v\xi^T\xi,$$

where $\xi(t) = \tilde{\xi} - k\widehat{L}_i^T e_i - kL_p^T\widehat{\Delta}e_{pd}$. Using the fact that $\widetilde{\Delta}(t) = \Delta - \widehat{\Delta}(t) = \begin{bmatrix}\tilde{d}^*(t)I_3 & 0_3\\ 0_3 & 0_3\end{bmatrix}$, the following substitutions can be made

$$e_i^T\widetilde{L}_i = e_i^T\widetilde{\Theta}_M L_{iM} = \tilde{\theta}^T e_{iM}L_{iM}$$
$$e_{pd}^T\widetilde{\Delta}L_pL_p^T\widehat{\Delta}e_{pd} = \tilde{d}^*\hat{d}^*x_d^Tx_d$$
$$\tilde{\xi}^TL_p^T\widetilde{\Delta}e_{pd} = \tilde{d}^*\tilde{\xi}_3^Tx_d$$
$$e_i^T\widehat{L}_iL_p\widetilde{\Delta}e_{pd} = \tilde{d}^*\left(e_i^T\widehat{L}_i\right)_3R_{vc}x_d$$

to yield

$$\dot{V} = -(k-k_vk^2)e_i^T\widehat{L}_i\widehat{L}_i^Te_i - k_\xi\tilde{\xi}^T\tilde{\xi} - (k-k_vk^2)e_{pd}^T\widehat{\Delta}L_pL_p^T\widehat{\Delta}e_{pd} - k_v\xi^T\xi - \tilde{\theta}^T\Gamma^{-1}\dot{\hat{\theta}} - \frac{1}{\gamma}\tilde{d}^*\dot{\hat{d}}^* - k\tilde{\theta}^Te_{iM}L_{iM}\widehat{L}_i^Te_i - k\tilde{d}^*\hat{d}^*x_d^Tx_d - k\tilde{d}^*\left(e_i^T\widehat{L}_i\right)_3R_{vc}x_d + \tilde{\theta}^Te_{iM}L_{iM}\tilde{\xi} - k\tilde{\theta}^Te_{iM}L_{iM}L_p^T\widehat{\Delta}e_{pd} + \tilde{d}^*\tilde{\xi}_3^Tx_d + 2k(k_vk-1)e_{pd}^T\widehat{\Delta}L_p\widehat{L}_i^Te_i. \quad (A.3)$$

After substituting the definitions for $\dot{\hat{\theta}}(t)$, $\dot{\hat{d}}^*(t)$ in (30) and (31) and eliminating terms, (A.3) can be reduced as

$$\dot{V} = -(k-k_vk^2)e_i^T\widehat{L}_i\widehat{L}_i^Te_i - k_\xi\tilde{\xi}^T\tilde{\xi} - (k-k_vk^2)e_{pd}^T\widehat{\Delta}L_pL_p^T\widehat{\Delta}e_{pd} - k_v\xi^T\xi + 2k(k_vk-1)e_{pd}^T\widehat{\Delta}L_p\widehat{L}_i^Te_i. \quad (A.4)$$

Completing squares for the terms in (A.4) yields (36).

Taking (A.4) and using the triangle inequality on the cross term $2e_{pd}^T\widehat{\Delta}L_p\widehat{L}_i^Te_i \leqslant \left\|\widehat{L}_i^Te_i\right\|^2 + \left\|L_p^T\widehat{\Delta}e_p\right\|^2$ yields

$$\dot{V} \leqslant -(k-k_vk^2)e_i^T\widehat{L}_i\widehat{L}_i^Te_i - k_\xi\tilde{\xi}^T\tilde{\xi} - (k-k_vk^2)e_{pd}^T\widehat{\Delta}L_pL_p^T\widehat{\Delta}e_{pd} - k_v\xi^T\xi - (k-k_vk^2)\left(\left\|\widehat{L}_i^Te_i\right\|^2 + \left\|L_p^T\widehat{\Delta}e_p\right\|^2\right) \quad (A.5)$$

Appendix B. Boundedness properties of $\dot{\hat{\Delta}}$, $\dot{L}_p$, $\dot{\widehat{L}}_i$ and $\dot{\widehat{L}}_i$

The matrix $\dot{\hat{\Delta}}(t)$ is given by

$$\dot{\hat{\Delta}} = \begin{bmatrix}\dot{\hat{d}}^*I_3 & 0_3\\ 0_3 & 0_3\end{bmatrix}. \quad (A.6)$$

If $\dot{\hat{d}}^*(t) \in \mathcal{L}_\infty$, then $\dot{\hat{\Delta}}(t) \in \mathcal{L}_\infty$.

The matrix $L_p(t) \in \mathbb{R}^{6\times6}$ is given in (13). A rotation matrix has a fixed norm, so $R_{vc}(t) \in \mathcal{L}_\infty$. The vector $u_\times(t)$ has a unit norm, so $u_\times(t) \in \mathcal{L}_\infty$. Exploiting the nonuniqueness of rotations, we map all values of $\varphi(t)$ to the range $\varphi(t) \in (-\pi,\pi]$. Thus $\varphi(t) \in \mathcal{L}_\infty$ and the singularity due to the $\text{sinc}^2\left(\frac{\varphi}{2}\right)$ term in the denominator is never encountered. Thus $L_p(t) \in \mathcal{L}_\infty$.

The matrix $\dot{L}_p(t) \in \mathbb{R}^{6\times6}$ is given by

$$\dot{L}_p = \begin{bmatrix}\dot{R}_{vc} & 0_{3\times3}\\ 0_{3\times3} & \dot{R}_{vc}L_\omega + R_{vc}\dot{L}_\omega\end{bmatrix} = \begin{bmatrix}\omega_{vc}^\times R_{vc} & 0_{3\times3}\\ 0_{3\times3} & \omega_{vc}^\times R_{vc}L_\omega + R_{vc}\dot{L}_\omega\end{bmatrix},$$

where $\omega_{vc}^\times(t) \in so(3)$ is the skew symmetric matrix form of the angular acceleration of the camera frame in the frame in which $\xi(t)$ is measured. Note that $\omega_{vc}^\times = 0_{3\times3}$ if the camera frame and input velocity frame are rigidly attached. The matrix $\dot{L}_\omega(t) \in \mathbb{R}^{3\times3}$ is given by

$$\dot{L}_\omega = -\frac{\dot{\varphi}}{2}u_\times - \frac{\varphi}{2}\dot{u}_\times - \frac{\dot{\varphi}u_\times^2}{1-\cos(\varphi)} + 2\left(1 - \frac{\text{sinc}(\varphi)}{\text{sinc}^2\left(\frac{\varphi}{2}\right)}\right)\dot{u}_\times u_\times,$$

which is singular at $\varphi(t) = 2k\pi, \forall k \in \mathbb{Z}/0$. If $\varphi(t) \neq 2k\pi$, and $\dot{e}_p(t) \in \mathcal{L}_\infty$, then $(u\dot{\varphi})(t) \in \mathcal{L}_\infty$ and $\dot{\varphi}(t), \dot{u}(t), \dot{L}_\omega(t) \in \mathcal{L}_\infty$. If $\omega_{vc}^\times(t) \in \mathcal{L}_\infty$ then $\dot{L}_p(t) \in \mathcal{L}_\infty$. If the velocity is defined in the camera frame, then $\omega_{vc}^\times = 0$ and is clearly bounded. The *image interaction matrix*, $L_i(t) \in \mathbb{R}^{2k\times6}$ is given in (16). In (16), $z_j(t)$ is the depth of the 3D point j in the camera frame and is assumed to be greater than some positive constant.

The derivative of $\widehat{L}_{ij}(t)$ from (16) is given in (A.7). The fact that $\dot{e}_i(t) \in \mathcal{L}_\infty$ implies that $\dot{u}_j(t), \dot{v}_j(t) \in \mathcal{L}_\infty$. Furthermore, $\dot{\alpha}_j(t) = -\dot{z}_j(t)\frac{z_j^*}{z_j^2(t)}$, and $\dot{e}_p(t) \in \mathcal{L}_\infty$ implies $\dot{z}_j(t) \in \mathcal{L}_\infty$. By assumption $z_j(t) > \epsilon > 0$. Thus $e_i(t), e_p(t), \dot{e}_i(t), \dot{e}_p(t), \hat{\theta}_j(t), \dot{\hat{\theta}}_j(t) \in \mathcal{L}_\infty$ is sufficient to show that $\dot{\widehat{L}}_i(t) \in \mathcal{L}_\infty$. Based on (34) and (36), $e_i(t), \tilde{\theta}(t), e_p(t), \tilde{d}(t), \tilde{\xi}(t) \in \mathcal{L}_\infty$ and $\left(L_p^T\widehat{\Delta}e_{pd} + \widehat{L}_i^Te_i\right), \tilde{\xi}(t), \xi(t) \in \mathcal{L}_2$. Since $\tilde{d}^*(t) \in \mathcal{L}_\infty$ and d^* is constant, it is clear that $\hat{d}^*(t), \widehat{\Delta}(t) \in \mathcal{L}_\infty$. Similarly, $\tilde{\theta}(t) \in \mathcal{L}_\infty$ implies $\hat{\theta}(t) \in \mathcal{L}_\infty$. As shown in above analysis, $L_p(t) \in \mathcal{L}_\infty$. By assumption, $\alpha_i(t) = \frac{z_i^*}{z_i(t)}$ is bounded since $z_i(t)$ is lower bounded by the physical size of the lens. So $e_i(t), \hat{\theta}(t), \alpha_i(t) \in \mathcal{L}_\infty$ imply $\widehat{L}_i \in \mathcal{L}_\infty$. The above results also show that $\xi_d(t) \in \mathcal{L}_\infty$.

The above analysis, along with (25) and (26) implies that $\dot{e}_p(t), \dot{e}_i(t) \in \mathcal{L}_\infty$, and $e_p(t)$, $e_i(t)$ are uniformly continuous. The above analysis and (30) and (31) imply that $\dot{\hat{\theta}}(t)$, $\dot{\hat{d}}^*(t), \dot{\widehat{\Delta}}(t) \in \mathcal{L}_\infty$, which implies $\widehat{\Delta}(t)$ is uniformly continuous. Since $\dot{e}_p(t) \in \mathcal{L}_\infty$ and $\omega_{vc}^\times(t) \in \mathcal{L}_\infty$ by assumption, Appendix B can be used to conclude that $\dot{L}_p(t) \in \mathcal{L}_\infty$ and $L_p(t)$ is uniformly continuous. If $e_p(t), \widehat{\Delta}(t)$ and $L_p(t)$ are uniformly continuous, then $\widehat{\Delta}(t)L_p(t)\Delta^{-1}e_p(t)$ is uniformly continuous. The fact that $\dot{e}_p(t), \dot{e}_i(t), \hat{\theta}(t), \dot{\hat{\theta}}(t) \in \mathcal{L}_\infty$ implies that $\dot{\widehat{L}}_i(t) \in \mathcal{L}_\infty$, which in turn implies that $\widehat{L}_i(t)$ and $\widehat{L}_i^T(t)e_i(t)$ is uniformly continuous. By assumption in Section 2, $\dot{J}(t)$, $\dot{q}(t) \in \mathcal{L}_\infty$. The previous results then show that $\bar{u}(t), \dot{\tilde{\xi}}(t) \in \mathcal{L}_\infty$, which means that $\tilde{\xi}(t)$ is uniformly continuous.

Appendix C. Proof that $\widehat{L}_i$ is full rank if and only if L_i is full rank

Lemma 1. *If a vector v is in the nullspace of $\widehat{L}_i(t)$, and $L_i(t)$ is full rank, then $v \neq [0,0,0,a,b,c]^T$, $a,b,c \in \mathbb{R}$*

$$\dot{\widehat{L}}_{ij} = \begin{bmatrix}\dot{\alpha}_j\hat{\theta}_j + \alpha_j\dot{\hat{\theta}}_j & 0 & u_j\dot{\alpha}_j\hat{\theta}_j + u_j\alpha_j\dot{\hat{\theta}}_j + \alpha_j\dot{u}_j\hat{\theta}_j & \cdots & -\dot{v}_ju_j - \dot{u}_jv_j & 2\dot{u}_ju_j & -\dot{v}_j\\ 0 & \dot{\alpha}_j\hat{\theta}_j + \alpha_j\dot{\hat{\theta}}_j & v_j\dot{\alpha}_j\hat{\theta}_j + v_j\alpha_j\dot{\hat{\theta}}_j + \alpha_j\dot{v}_j\hat{\theta}_j & \cdots & -2\dot{v}_jv_j & \dot{v}_ju_j + \dot{u}_jv_j & \dot{u}_j\end{bmatrix} \quad (A.7)$$

Proof. Proceed with proof by contradiction. Assume $L_i(t)$ is full rank, $\widehat{L}_i(t)$ is not full rank and $v = [0,0,0,a,b,c]^T$ is in the nullspace of $\widehat{L}_i(t)$. Then v is in the nullspace of $L_i(t)$, because they have the same right three columns. However, $L_i(t)$ is full rank which is a contradiction so $v \neq [0, 0, 0, a, b, c]^T, a, b, c \in \mathbb{R}$. □

Lemma 2. *If a vector v is in the nullspace of $L_i(t)$, and $\widehat{L}_i(t)$ is full rank, then $v \neq [0,0,0,a,b,c]^T, a, b, c \in \mathbb{R}$.*

Proof. See the proof for Lemma 1. □

Theorem. *:If $L_i(t)$ is full rank, and no $\theta_j = 0$, $j \in \{1,2,3\}$, then $\widehat{L}_i(t)$ is full rank. Similarly, if $\widehat{L}_i(t)$ is full rank, and no $\theta_j = 0, j \in \{1,2,3\}$, then $L_i(t)$ is full rank.*

Proof. Proceed with proof by contradiction. Assume $L_i(t)$ is full rank, $\nexists \theta_j = 0$, $j \in \{1,2,3\}$, and $\widehat{L}_i(t)$ is not full rank. Taking SVD of $\widehat{L}_i(t)$ gives

$$\widehat{L}_i U \Sigma V^T = L_i - \widetilde{L}_i$$
$$U\Sigma = L_i V - \widetilde{L}_i V,$$

where U and V are full rank, orthonormal matrices.

If $\widehat{L}_i(t)$ is not full rank, the sixth singular value is 0, i.e., the (6,6) element of Σ is 0. This implies that the sixth column of $U\Sigma = 0$, and

$$L_i V_6 = \widetilde{L}_i V_6,$$

where V_6 is the sixth column of V, and by assumption $L_i(t)$ is full rank. Furthermore, since the right three columns of $\widetilde{L}_i(t)$ are all zeros, we can rewrite this as

$$L_{i,3} V_{6,3} = \widetilde{L}_{i,3} V_{6,3}, \tag{A.8}$$

where $L_{i,3}(t)$ is the first three columns of $L_i(t)$, $\widetilde{L}_{i,3}(t)$ is the first three columns of $\widetilde{L}_i(t)$ and $V_{6,3}$ is the first elements of V_6.

If $\widehat{L}_i(t)$ is not full rank, then V_6 is in the nullspace of $\widehat{L}_i(t)$. Since $L_i(t)$ is full rank $V_{6,3} \neq [0,0,0]^T$ by Lemma 1, i.e., at least one element of $V_{6,3}$ is nonzero. If $V_{6,3} = [a,b,c], a, b \in \mathbb{R}/0, c \in \mathbb{R}$ (i.e., the first and/or second elements of $V_{6,3}$ are not 0), then it is seen from (16)–(18) and (A.8) that

$$\sum_j \alpha_j \theta_j = \sum_j \alpha_j \tilde{\theta}_j$$
$$\sum_j \alpha_j \hat{\theta}_j = 0. \tag{A.9}$$

The facts that $\alpha_j(t) > 0$, and (A.9) is true if and only if $\forall_j \hat{\theta}_j(t) = 0$ lead to a contradiction the assumptions. If $V_{6,3} = [0,0,c], c \in \mathbb{R}/0$ (i.e., the first and second elements are 0), then

$$\sum_j (u_j + v_j)\alpha_j \hat{\theta}_j = 0. \tag{A.10}$$

By assumption it is not true that $\forall_j (u_j + v_j) = 0$, so (A.10) is true if and only if $\forall_j \hat{\theta}_j(t) = 0$, which contradicts the assumptions. By following the above argument, it can be proven that if $\widehat{L}_i(t)$ is full rank, and no $\theta_j = 0, j \in \{1,2,3\}$, then $L_i(t)$ is full rank. □

References

[1] Espiau B, Chaumette F, Rives P. A new approach to visual servoing in robotics. IEEE Trans Robot Automat 1992;8(3):313–26.

[2] Martinet P, Gallice J, Khadraoui D. Vision based control law using 3D visual features. In: Proceedings of WAC 96, vol. 3; 1996. p. 497–502.

[3] Hutchinson S, Hager G, Corke P. A tutorial on visual servo control. IEEE Trans Robot Automat 1996;12(5):651–70.

[4] Chaumette F. Potential problems stability and convergence in image-based and position-based visual servoing. In: Kriegman D, Hager G, Morse S, editors. The confluence vision and control. Lecture notes in control and information sciences, vol. 237. Springer-Verlag; 1998. p. 66–78.

[5] Malis E, Chaumette F, Boudet S. 2-1/2D visual servoing. IEEE Trans Robot Automat 1999;15(2):238–50.

[6] Deguchi K. Optimal motion control for image-based visual servoing by decoupling translation and rotation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems; 1998. p. 705–11.

[7] Malis E, Chaumette F. 2 1/2D visual servoing with respect to unknown objects through a new estimation scheme camera displacement. Int J Comput Vis 2000;37(1):79–97.

[8] Corke P, Hutchinson S. A new partitioned approach to image-based visual servo control. IEEE Trans Robot Automat 2001;17(4):507–15.

[9] Chen J, Dawson DM, Dixon WE, Behal A. Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension. IEEE Trans Control Systems Technol 2005;13(5):814–25.

[10] Hu G, Gans N, Fitz-Coy N, Dixon W. Adaptive homography-based visual servo tracking control via a quaternion formulation. IEEE Trans on Control Systems Technol 2010;18(1):128–35.

[11] Faugeras OD, Lustman F. Motion and structure from motion in a piecewise planar environment. Int J Pattern Recog Artificial Intell 1988;2(3):485–508.

[12] Gans N, Hutchinson S. Stable visual servoing through hybrid switched-system control. IEEE Trans Robot 2007;23(3):530–40.

[13] Chesi G, Hashimoto K, Prattichizzo D, Vicino A. Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. IEEE Trans Robot 2004;20(5):908–14.

[14] Deng L, Janabi-Sharifi F, Wilson W. Hybrid motion control and planning strategies for visual servoing. IEEE Trans Indust Eng 2005;52(4):1024–40.

[15] Cowan N, Weingarten J, Koditschek D. Visual servoing via navigation functions. IEEE Trans Robot Automat 2002;18(4):521–33.

[16] Chitrakaran V, Dawson DM, Dixon WE, Chen J. Identification a moving object's velocity with a fixed camera. Automatica 2005;41(3):553–62.

[17] Lapreste J, Mezouar Y. A hessian approach to visual servoing. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems; 2004. p. 998–1003.

[18] Fomena R, Chaumette F. Improvements on visual servoing from spherical targets using a spherical projection model. IEEE Trans Robotics 2009;25(4):874–86.

[19] Tahri O, Mezouar Y, Chaumette F, Corke P. Decoupled image-based visual servoing for cameras obeying the unified projection model. IEEE Trans Robotics 2010;26(4):684–97.

[20] DeMenthon D, Davis LS. Model-based object pose in 25 lines code. In: European conference on computer vision; 1992. p. 335–43.

[21] Hartley R. In defence of the eight-point algorithm. IEEE Trans Pattern Anal Machine Intell 1997;19:580–93.

[22] Fang Y, Dixon WE, Dawson DM, Chawda P. Homography-based visual servo regulation of mobile robots. IEEE Trans Syst Man Cybern 2005;35(5):1041–50.

[23] Cheah C, Liu C, Slotine J. Adaptive vision based tracking control of robots with uncertainty in depth information. In: Proceedings of the IEEE international conference robotics and automation; 2007. p. 2817–22.

[24] De Luca A, Oriolo G, Giordano P. On-line estimation of feature depth for image-based visual servoing schemes. In: Proceedings of the IEEE international conference robotics and automation; 2007. p. 2823–8.

[25] Wang H, Liu YH, Zhou D. Adaptive visual servoing using point and line features with an uncalibrated eye-in-hand camera. IEEE Trans Robotics 2008;24(4):843–57.

[26] Cheah CC, Liu C, Slotine JJE. Adaptive jacobian vision based control for robots with uncertain depth information. Automatica 2010;46(7):1228–33. doi:10.1016/j.automatica.2010.04.009.

[27] Hu G, Gans N, Dixon WE. Complexity and nonlinearity in autonomous robotics, encyclopedia of complexity and system science, chap. Adaptive Visual Servo Control, vol. 1. Springer; 2009. p. 42–63.

[28] Hafez A, Jawahar C. Visual servoing by optimization of a 2d/3d hybrid objective function. In: Proceedings of the IEEE international conference robotics and automation; 2007. p. 1691–6.

[29] Dixon WE, Behal A, Dawson DM, Nagarkatti SP. Nonlinear control of engineering systems, a Lyapunov-based approach. Birkhauser; 2003.

[30] Kelly R, Carelli R, Nasisi O, Kuchen B, Reyes F. Stable visual servoing camera-in-hand robotic systems. IEEE/ASME Trans Mechatron 2000;5(1):39–48.

[31] Nasisi O, Carelli R. Adaptive servo visual robot control. Robot Auto Syst 2003;43(1):51–78.

[32] Ma Y, Soatto S, Koseck J, Sastry S. An Invitation to 3-D Vision. Springer; 2004.

[33] Zhang Z, Hanson A. 3D reconstruction based on homography mapping. In: Proceedings of the ARPA image understanding workshop palm. Springs CA; 1996.

[34] Longuet-Higgins H. A computer algorithm for reconstructing a scene from two projections. Nature 1981;293:133–5.
[35] Quan L, Lan ZD. Linear *n*-point camera pose determination. IEEE Trans Pattern Anal Machine Intell 1999;21(8):774–80.
[36] Weiss LE, Sanderson AC, Neuman CP. Dynamic sensor-based control robots with visual feedback. IEEE Trans Robot Automat 1987;3(5):404–17.
[37] Slotine J, Li W. Applied nonlinear control. Prentice Hall; 1991.
[38] Tsai R, Lenz R. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. IEEE Trans Robot Auto 1989;5(3):345–58.
[39] Silveira G, Malis E. Direct visual servoing with respect to rigid objects. In: International conference on intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ; 2007. p. 1963–8.