Acceleration Command based Visual Servoing with Artificial Induced Time-Delay

Madhu Yadav
madhuyadav@ee.iitd.ac.in
Control & Automation Group
Department of Electrical Engineering
Indian Institute of Technology Delhi
New Delhi, India

Chimmula Kishore
chimmulakishore@ee.iitd.ac.in
Control & Automation Group
Department of Electrical Engineering
Indian Institute of Technology Delhi
New Delhi, India

Indra Narayan Kar
ink@ee.iitd.ac.in
Control & Automation Group
Department of Electrical Engineering
Indian Institute of Technology Delhi
New Delhi, India

Sumantra Dutta Roy
sumantra@ee.iitd.ac.in
Computer Technology Group
Department of Electrical Engineering
Indian Institute of Technology Delhi
New Delhi, India

ABSTRACT

The tracking of an object using visual feedback by controlling the camera motion is known as Visual Servoing. In general most of visual servoing techniques use camera velocity as control input. In this paper, camera acceleration is used for deriving the control law. The use of acceleration input can accomplish smoother operating trajectory. However, this controller requires the first order feature measurement which may not be available. In this paper, to circumvent the issue, a time-delayed approximation is used for the non-measurable system states. The controller parameters are obtained from the stability analysis of the closed-loop time-delay system using Lyapunov-Razumikhin approach. The implemented controller is able to track the desired feature smoothly. These results establish a complementary approach to the existing acceleration-based augmented controller design.

CCS CONCEPTS

• **Computing methodologies** → **Vision for robotics**; • **Computer systems organization** → *Robotic control.*

KEYWORDS

Visual Servoing, Time-delayed Control, Lyapunov-Razumikhin, Acceleration command

1 INTRODUCTION

In engineering, tracking of an object using visual feedback information obtained from a camera to control the motion of a robot is known as vision based control or visual servoing [1]. Such control system broadly categories into two groups, position-based visual servoing (PBVS) and image-based visual servoing (IBVS). PBVS uses the error which is a function of Cartesian co-ordinates of features from an object in the world to design the control law. This approach has an advantage that it directly controls the camera trajectory in Cartesian space assuming 3D model of the object is known. But if there exist an error, while calibrating camera or getting 3D model of the target, this may lead the error to propagate and results in the feature points to leave the field of view. Whereas IBVS uses the error which is directly expressed in terms of image features only. This approach is considered more flexible than PBVS as it does not require explicit calculation of the target pose from image features. But the error in calibration, might result in performance degradation in case of both IBVS and PBVS. Apart from IBVS and PBVS, there exist other servoing techniques in visual servoing like hybrid visual servoing which combines the advantages of both IBVS and PBVS [2]. In this paper, an IBVS based design for generating a control input has been proposed.

In general, most of the controllers available in visual servoing literature uses the camera velocity as control input, similar to the kinematic control of a robot [3]. Vision based controllers derived based on the first order dynamics of the image features often use feedback of image feature locations to generate the control action in terms of camera velocity. However, these velocity based control input may not result in smooth trajectories while achieving tracking objective for certain operating conditions. Velocity based controller design can allow us only to design a proportional controller, which has effects on overshoot and smoothness of trajectory. The use of acceleration based control signal can enhance the smoothness of the trajectory sufficiently. Furthermore, these acceleration based designs serve as counterpart to the torque based dynamic controller design for a robot. The controller design incorporating dynamics instead of kinematics which results in smoother performance is a well known fact in robotics community [4]. Though such acceleration based feedback can perform better in aspects of smoothness of trajectory, it requires feedback of first order feature error (i.e., velocity feedback), which may not be straight forward to measure from the image information. To circumvent this issue,

©2019 AIR'19, July 2019, Chennai, India
https://doi.org/xxxxxxxxx...$15.00

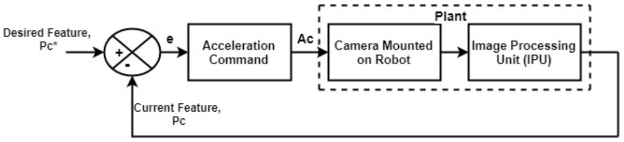

Figure 1: Block Diagram Representation of Control Architecture

the non-measurable first order feature error derivative is approximated with Euler model, which is popularly known as time-delayed control. Time-delayed control(TDC) have been used efficiently in robotics [5, 6], smartgrid [7], process industry [8]. TDC provides a linear time-delayed approximation in place of state derivative which is not measurable. In this particular case it is the derivative of feature position.

In this article, the main contribution is towards approximating the error dynamics with linear artificial time-delayed model to avoid non-measurable states to be used in control input. This article has three major sections. Section I talks about the existing results and the literature in this domain. Section II majorly focuses on system modeling and controller design using Lyapunov based methods. The section III presents simulation results for a case study.

2 DYNAMICS & CONTROLLER DESIGN

In this section, the feature dynamics are formulated using the geometry of perspective projection and these are used later on for designing the acceleration command used in visual servoing along with adequate stability analysis.

2.1 Feature Dynamics

The projection of a 3D point, P_w located on a target object in Cartesian space on to the 2D image space is done by a camera [9] through perspective projection methods. The location of this projected point within the image space is treated as a feature. This feature can be located inside the image using pixel coordinates/image space coordinates with respect to image reference frame.

Using pin-hole camera model [10], a relation between world point P_w with Cartesian coordinates $P_w = [X_w, Y_w, Z_w]^T$ and its projection p_c with normalized image coordinates $p_c = [x_c, y_c]^T$ can be obtained as in (1)

$$\begin{aligned} x_c &= \frac{X_w}{Z_w} \\ y_c &= \frac{Y_w}{Z_w} \end{aligned} \tag{1}$$

Image space coordinates (which can be directly measured from the image) $[u_c,\ v_c]^T$ of the feature can be calculated from the normalized image coordinates by a mathematical relation given as

$$\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = K_c^{-1} \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} \tag{2}$$

where, intrinsic camera parameter matrix K_c is defined as,

$$K_c = \begin{bmatrix} f_x & \mu & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The parameters, f_x and f_y are scale factors in u and v direction in image reference frame and are proportional to the focal length, $(u_0,\, v_0)$ is the principle point of the camera and μ is skew coefficient. These constants $f_x,\ f_y, \mu,\ u_0$ and v_0 are called as intrinsic parameters of the camera, can be estimated using camera calibration methods [11].

Using (1), the dynamics of the feature p_c can be written as [12].

$$\dot{p}_c = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} = \begin{bmatrix} \frac{\dot{X}_w}{Z_w} - \frac{X_w \dot{Z}_w}{Z_w^2} \\ \frac{\dot{Y}_w}{Z_w} - \frac{Y_w \dot{Z}_w}{Z_w^2} \end{bmatrix} \tag{4}$$

Further, this can be simplified using (1) as

$$\dot{p}_c = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} = \begin{bmatrix} \frac{\dot{X}_w - x_c \dot{Z}_w}{Z_w} \\ \frac{\dot{Y}_w - y_c \dot{Z}_w}{Z_w} \end{bmatrix} \tag{5}$$

The second order dynamics of the feature p_c can be obtained from (4) as

$$\ddot{p}_c = \begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \end{bmatrix} = \begin{bmatrix} \frac{\ddot{X}_w}{Z_w} - \frac{X_w \ddot{Z}_w}{Z_w^2} - 2\frac{\dot{X}_w \dot{Z}_w}{Z_w^2} + 2\frac{X_w {\dot{Z}_w}^2}{Z_w^3} \\ \frac{\ddot{Y}_w}{Z_w} - \frac{Y_w \ddot{Z}_w}{Z_w^2} - 2\frac{\dot{Y}_w \dot{Z}_w}{Z_w^2} + 2\frac{Y_w \dot{Z}_w^2}{Z_w^3} \end{bmatrix} \tag{6}$$

Using the concepts of rigid body motion [13], the dynamics of a point moving with linear velocity, $v \in \mathbb{R}^3$ and angular velocity, $\omega \in \mathbb{R}^3$ can be given as in (7). Furthermore, taking the second derivative for the same dynamics with linear acceleration, $a_c \in \mathbb{R}^3$ and angular acceleration $\phi_c \in \mathbb{R}^3$ can be represented as

$$\dot{P}_w = -v - \omega \times P_w \tag{7}$$

$$\ddot{P}_w = -a - \phi \times P_w + 2\omega \times v + \omega \times (\omega \times P_w) \tag{8}$$

These dynamics are also valid for a stationary point observed by a moving camera with velocity, $V_c = [v_c, \omega_c]^T$ where $v_c \in \mathbb{R}^3$ and $\omega_c \in \mathbb{R}^3$ are the linear and angular velocity vector components respectively.

Using (5) & (7), the first-order dynamics of p_c as in [12] can be obtained as

$$\dot{p}_c = J_a V_c \tag{9}$$

where J_a is camera interaction matrix. The Interaction matrix for a feature point p_c is expressed in similar fashion to the classical IBVS technique [1]. The expression for J_a can be given as

$$\mathbf{J}_a = \begin{bmatrix} -\frac{1}{Z_w} & 0 & \frac{x_c}{Z_w} & (x_c y_c) & -(1 + x_c^2) & y_c \\ 0 & -\frac{1}{Z_w} & \frac{y_c}{Z_w} & (1 + y_c^2) & -(x_c y_c) & -x_c \end{bmatrix} \tag{10}$$

where Z_w is depth of the world point from the camera frame.

Similarly using (6) & (8), the second-order dynamics for p_c as in [4] can be written as

$$\ddot{p}_c = J_a A_c + J_v \tag{11}$$

where $A_c \in \mathbb{R}^6$ is the acceleration command for the camera, and J_v is obtained by using (12)

$$J_v = \begin{bmatrix} {V_c}^T \beta_x V_c \\ {V_c}^T \beta_y V_c \end{bmatrix} \tag{12}$$

Expressions taken from [4] for β_x and β_y are given in equation (13) and (14).

$$\beta_x = \begin{bmatrix} 0 & 0 & \frac{1}{Z_w^2} & \frac{-y_c}{Z_w} & \frac{3x_c}{2Z_w} & 0 \\ 0 & 0 & 0 & \frac{-x_c}{2Z_w} & 0 & \frac{-1}{2Z_w} \\ \frac{1}{Z_w^2} & 0 & \frac{2x_c}{Z_w^2} & \frac{2x_c y_c}{Z_w} & -\frac{1}{2Z_w} - \frac{2x_c^2}{Z_w} & \frac{y_c}{Z_w} \\ \frac{-y_c}{Z_w} & \frac{-x_c}{2Z_w} & \frac{2x_c y_c}{Z_w} & x_c + 2x_c y_c^2 & \frac{-y_c}{2} - 2x_c^2 y_c & \frac{1}{2} - \frac{x_c^2}{2} + y_c^2 \\ \frac{3x_c}{2Z_w} & 0 & \frac{-1}{2Z_w} - \frac{2x_c^2}{Z_w} & -\frac{y_c}{2} - 2x_c^2 y_c & 2x_c + 2x_c^3 & -\frac{3x_c y_c}{2} \\ 0 & \frac{-1}{2Z_w} & \frac{y}{Z_w} & \frac{1}{2} - \frac{x_c^2}{2} + y_c^2 & -\frac{3x_c y_c}{2} & -x_c \end{bmatrix} \tag{13}$$

$$\beta_y = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{2y_c}{Z_w} & \frac{1}{2Z_w} \\ 0 & 0 & \frac{1}{Z_w^2} & -\frac{3y_c}{2Z_w} & \frac{x_c}{Z_w} & 0 \\ 0 & \frac{1}{Z_w^2} & \frac{2y_c}{Z_w^2} & \frac{1}{2Z_w} + \frac{y_c^2}{Z_w} & -\frac{x_c y_c}{Z_w} & -\frac{x_c}{Z_w} \\ 0 & -\frac{3y_c}{2Z_w} & -\frac{1}{2Z_w} + \frac{y_c^2}{Z_w} & 2y_c + 2y_c^3 & -\frac{x_c}{2} - 2x_c y_c^2 & -\frac{3x_c y_c}{2} \\ \frac{1}{2Z_w} & \frac{x_c}{y_c} & -\frac{x_c y_c}{Z_w} & -\frac{x_c}{2} - 2x_c y_c^2 & y_c + 2x_c^2 y_c & \frac{1}{2} + x_c^2 - \frac{y_c^2}{2} \\ \frac{1}{2Z_w} & 0 & -\frac{x_c}{Z_w} & \frac{3x_c y_c}{2} & \frac{1}{2} + x_c^2 - \frac{y_c^2}{2} & -y_c \end{bmatrix} \tag{14}$$

2.2 Controller Design

The objective of a vision based control scheme is to track the desired image feature [14]. Error between the desired image feature location p_c^*, which is assumed as a constant and current image feature location $p_c(t)$ can be represented as

$$e(t) = p_c(t) - p_c^* \tag{15}$$

Using (9) and (11), the error dynamics can be written as,

$$\begin{aligned} \dot{e}(t) &= \dot{p}_c(t) = J_a V_c \\ \ddot{e}(t) &= \ddot{p}_c(t) = J_a A_c + J_v \end{aligned} \tag{16}$$

The second order error dynamics in (16) has acceleration command A_c which has to be designed such that the error converges to zero with time. Generally, the vision based systems are controlled with velocity input which has undesirable characteristics like enhancing the feedback sensor noise which results in non-smoother trajectory [15]. Here, an acceleration based design as in (17) is helpful to suppress these effects and also stabilize the error dynamics, such that $e(t)$ converges to zero over time.

$$A_c = {J_a}^{\dagger}[-J_v - k_a \dot{e} - e] \tag{17}$$

where $k_a > 0$ and ${J_a}^{\dagger}$ is pseudo-inverse of interaction matrix [16]. The closed-loop error dynamics for acceleration input as in (17) can be given as

$$\ddot{e} = J_a J_a^{\dagger}(-J_v - k_a \dot{e} - e) \tag{18}$$

$$\ddot{e} = -k_a \dot{e} - e + (I - J_a J_a^{\dagger}) J_v \tag{19}$$

where I is an identity matrix.

By proper selection of desired image features from an image, we can make J_a full rank [1]. In this work, it is assumed that the selected features lie on a planar surface in the world/Cartesian reference frame in a square shape pattern. As $J_a J_a^{\dagger} = (J_a J_a^T)\,(J_a J_a^T)^{-1} \approx I$, we will obtain

$$\ddot{e} \approx -k_a \dot{e} - e \tag{20}$$

The effect of this approximation is assumed to be negligible for this work. This approximation error can be included in the analysis by clubbing it with the time delayed approximation error. The control law in (17) requires the feedback of feature error derivative which is not directly measurable from the image data. As the derivative of image feature locations are not directly measurable from the vision sensor, the first order derivative of image feature location error is replaced with its time-delayed approximation (this is done by using the past data i.e., the difference between current image feature location and the previous image feature location over a finite time interval) by introducing an artificial delay expressed as below

$$\ddot{e} = -k_a \frac{(e - e_h)}{h} - e \tag{21}$$

$$\ddot{e} = -\left(\frac{k_a}{h} + 1\right) e + \frac{k_a}{h} e_h \tag{22}$$

where $e_h = e(t - h)$ is the past instant value of the error and h is a constant artificial delay introduced into the system.
The time-delayed approximated acceleration command can be written from (17) as

$$A_c = {J_a}^{\dagger}[-J_v - k_a \frac{(e - e_h)}{h} - e] \tag{23}$$

Considering,

$$\begin{aligned} x_1(t) &= e(t) \\ x_2(t) &= \dot{e}(t) \end{aligned} \tag{24}$$

Therefore, the state-space representation for the error dynamics can be given as

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= -\left(\frac{k_a}{h} + 1\right) x_1 + \frac{k_a}{h} x_1(t - h) \end{aligned} \tag{25}$$

Matrix form representation of (25) with $\alpha = \left(\frac{k_a}{h} + 1\right)$ is as follows

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\alpha & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ (\alpha - 1) & 0 \end{bmatrix} \begin{bmatrix} x_{1h} \\ x_{2h} \end{bmatrix} \tag{26}$$

$$\implies \dot{x}(t) = Ax(t) + Bx(t - h) \tag{27}$$

where, $x(t) = [\,x_1(t) \quad x_2(t)\,]^T$, $x_h(t) = x(t-h) = [\,x_1(t-h) \quad x_2(t-h)\,]^T$, $A = \begin{bmatrix} 0 & 1 \\ -\alpha & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 & 0 \\ (\alpha - 1) & 0 \end{bmatrix}$.

Here, α is a tunable design parameter. It is interesting to note that as time delay h increases, k_a also need to be increased to maintain the required α. The effect of varying h and k_a on the system performance has been discussed in the coming section with the help of simulations.

Stability Analysis. Although the acceleration command shown in (17), ensures closed-loop stability but as $\dot{e}$ is not measurable, the approximated closed-loop error dynamics in (22) has time-delayed component in it, due to which the stability may not be conserved from (17). The stability of time-delayed systems can be explained using Lyapunov-Razumikhin condition [17]. The stability procedure followed in [7], [18] has been adopted here. The stability proof is along the line of [18].

For the selection of a candidate Lyapunov function $V(x) = \frac{1}{2}x^T Px$, it is assumed that $V(x)$ satisfies (28) as in Razumikhin-type theorem, for a constant $\zeta > 1$.

$$V(x(\tau)) \leq \zeta V(x(t)) \qquad t - 2h \leq \tau \leq t \tag{28}$$

Taking time derivative of V results in

$$\dot{V} = \dot{x}^T Px + x^T P\dot{x} \tag{29}$$

where $P > 0$ is the solution of Lyapunov equation $A^T P + PA = -Q$ for some $Q > 0$.
Now, using (27), the equation (29) can be represented as,

$$\dot{V} = [Ax + Bx_h]^T Px + x^T P[Ax + Bx_h] \tag{30}$$

where x_h can be written in the integral form as

$$x(t-h) = x_h(t) = x(t) - \int_{-h}^{0} \dot{x}(t+\theta)d\theta \tag{31}$$

Substituting (31) in (30) results in

$$\begin{aligned} \dot{V} =& \left[Ax + B\left(x(t) - \int_{-h}^{0} \dot{x}(t+\theta)d\theta\right)\right]^T Px \\ &+ x^T P\left[Ax + B\left(x(t) - \int_{-h}^{0} \dot{x}(t+\theta)d\theta\right)\right] \end{aligned} \tag{32}$$

Simplifying and rewriting the terms in (32) yields

$$\begin{aligned} \dot{V} =& x^T \left[A^T P + B^T P + PA + PB\right] x \\ &- \left(\int_{-h}^{0} \dot{x}^T(t+\theta)d\theta\right) B^T Px - x^T PB\left(\int_{-h}^{0} \dot{x}(t+\theta)d\theta\right) \end{aligned} \tag{33}$$

$$\dot{V} = x^T[A^T P + B^T P + PA + PB]x - 2x^T PB\int_{-h}^{0} \dot{x}(t+\theta)\, d\theta \tag{34}$$

Now, expanding the second term in (34) using (27) gives

$$-2x^T PB\int_{-h}^{0} \dot{x}(t+\theta)d\theta = -2x^T PB\int_{-h}^{0} [Ax(t+\theta) + Bx(t+\theta-h)]d\theta \tag{35}$$

By substituting (35) into (34), we get

$$\begin{aligned} \dot{V} =& x^T[A^T P + B^T P + PA + PB]x \\ &- \underbrace{2x^T PBA\int_{-h}^{0} x(t+\theta)d\theta}_{I_1} - \underbrace{2x^T PBB\int_{-h}^{0} x(t+\theta-h)]d\theta}_{I_2} \end{aligned} \tag{36}$$

The term, I_1 in the equation (36) can be further simplified as,

$$I_1 = 2x^T PBA\int_{-h}^{0} x(t+\theta)d\theta \tag{37}$$

$$= -\int_{0}^{-h} \underbrace{2x^T PBA}_{2w_1^T}\ \underbrace{x(t+\theta)}_{w_2}\ d\theta \tag{38}$$

Using the inequality shown in (39) for any two non-zero vectors w_1 and w_2 with a constant $\gamma > 0$ and a positive definite matrix $M > 0$, we can write a bound for (37) as in (40) by replacing $M = P$ in (39).

$$-2{w_1}^T w_2 \leq \gamma {w_1}^T M^{-1} w_1 + \frac{1}{\gamma}{w_2}^T M w_2 \tag{39}$$

$$I_1 \leq \int_{0}^{-h} \left(\gamma x^T PBAP^{-1}A^T B^T Px \ + \ \frac{1}{\gamma}[x^T(t+\theta)Px(t+\theta)]\right) d\theta \tag{40}$$

Simplifying the equation (40),

$$I_1 \leq \int_{0}^{-h} [\gamma x^T PBAP^{-1}A^T B^T Px]d\theta + \int_{0}^{-h} [\frac{1}{\gamma}[x^T(t+\theta)Px(t+\theta)]d\theta \tag{41}$$

$$= -h\gamma x^T PBAP^{-1}A^T B^T Px + \int_{0}^{-h} \frac{1}{\gamma}[x^T(t+\theta)Px(t+\theta)]d\theta \tag{42}$$

Using equation (28), we can further bound (42) as follows

$$I_1 \leq -h\gamma x^T PBAP^{-1}A^T B^T Px + \int_{0}^{-h} \frac{\zeta}{\gamma}[x^T Px]d\theta \tag{43}$$

$$= -h\gamma x^T PBAP^{-1}A^T B^T Px - h\frac{\zeta}{\gamma}[x^T Px] \tag{44}$$

$$\implies I_1 \leq -\left(hx^T[\gamma PBAP^{-1}A^T B^T P + \frac{\zeta}{\gamma}P]x\right) \tag{45}$$

Similar to I_1, the term I_2 can be bounded as,

$$I_2 = 2x^T PBB\int_{-h}^{0} x(t+\theta-h)d\theta \tag{46}$$

$$\implies I_2 \leq -\left(hx^T[\gamma PBBP^{-1}B^T B^T P + \frac{\zeta}{\gamma}P]x\right) \tag{47}$$

Substituting the final bounds of I_1 and I_2 back into the equation (36) gives

$$\begin{aligned} \dot{V} \leq& x^T[A^T P + B^T P + PA + PB + h\gamma PBAP^{-1}A^T B^T P + h\frac{\zeta}{\gamma}P \\ &+ h\gamma PBBP^{-1}B^T B^T P + h\frac{\zeta}{\gamma}P]x \end{aligned} \tag{48}$$

$$\begin{aligned} =& x^T[h\left(\gamma PBAP^{-1}A^T B^T P + \gamma PBBP^{-1}B^T B^T P + 2\frac{\zeta}{\gamma}P\right) \\ &+ A^T P + PA + B^T P + PB]x \end{aligned} \tag{49}$$

$$\implies \dot{V} \leq x^T[hE - cQ]x \tag{50}$$

where, $E = [\gamma PBAP^{-1}A^T B^T P + \gamma PBBP^{-1}B^T B^T P + 2\frac{\zeta}{\gamma}P]$, $A^T P + PA + B^T P + PB = -c\ Q$, $Q > 0$ and c is a positive constant which can regulate the transient response of the system.

In order to show stability of the closed-loop approximated time delayed system in (27) using Lyapunov stability theorem, it is required that $\dot{V} \leq 0$. From (50), this can be made possible only if $(hE - cQ)$ is a negative definite matrix. To make $(hE - cQ)$ negative definite we can obtain an upper bound on the maximum delay that we can put into the system as follows

$$h < (c\ \lambda_{min}(Q))/||E|| \tag{51}$$

where $\lambda_{min}(Q)$ represents the minimum eigen value of positive definite matrix Q. Equation (51) provides a sufficient condition for stability in presence of time-delay. It is evident from (50) that when delay h is zero (i.e., delay free system), the system (27) is asymptotically stable.

3 SIMULATION RESULTS

In this section, the proposed Time-Delayed Acceleration Controller (TDAC) is validated with a case study of camera mounted at the end-effector of a 6 DoF manipulator. The block diagram representation of the closed loop system is shown in Figure 1. As camera is rigidly mounted at the end-effector, the acceleration and velocity of the camera is same as the acceleration and velocity of the end-effector. Once we design a controller for controlling the motion of camera/end-effector, the same controller can be transformed into joint space using the manipulator Jacobian. So, from now onwards we only deal with controlling camera motion. The system is modeled as per (9) and (11), for which the camera parameters such as focal length and the skew coefficient are chosen as $f = 0.1$, $\mu = 0$ and principal point (u_0, v_0) is assumed to be at the center of the image.

For this system, a set of four points forming a square is considered as the desired feature. The goal of the controller is to take these features from the current location to desired location $s^* = [(-55, -45), (55, -45), (55, 45), (-55, 45)]$ by controlling the motion of camera.

The closed-loop system performance depends on the selection of the parameter $\alpha = (1 + \frac{k_a}{h})$. α can be obtained from the algebraic relation for a fixed value of delay h given in (51). For the value of $\alpha = 0.1$ and $h = 0.01$, the closed-loop system performance is shown in Figure (2). Where, the system is able to track to the desired feature location (magenta box) from an arbitrary initial feature location (blue box). The individual feature can be traced with the line connecting the nodes.

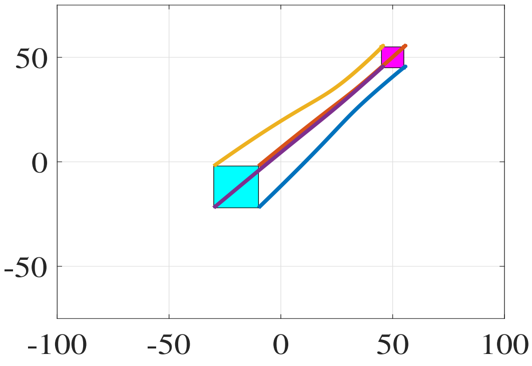

Figure 2: The feature traces to the desired feature for designed controller in (23)

As the states reaches to desired steady-state, the value of $e(t)$ and $e(t - h)$ goes to zero, hence from (23) the control effort goes to zero. But, in case of ideal derivative based feedback $e(t) = 0$ can also make $\dot{e}(t) \neq 0$ which can generate unnecessary control effort. As the controller takes account of error of present and past, it is intuitive that large delay can raise oscillations or may destabilize the system. To understand this, the closed-loop system is simulated with multiple time-delay values, unlike the values selected based on the design conditions in (51). The simulation results for error performance for multiple delays is plotted in Figure 3. The plot shows the error for one of the feature points with respect to time for visualizing the transient response. From Figure 3, it can be seen that the error reduces to zero for $h = 0.01$ to 0.1 but as the delay increases, we can observe an oscillation for $h = 0.1$. Also, for larger delay such as $h = 0.5$ the feature error becomes unstable and faced singularity issue. So, it can be inferred that on increasing the values of delay, the transient response becomes faster but may raise oscillation and for a significant delay the system may become unstable.

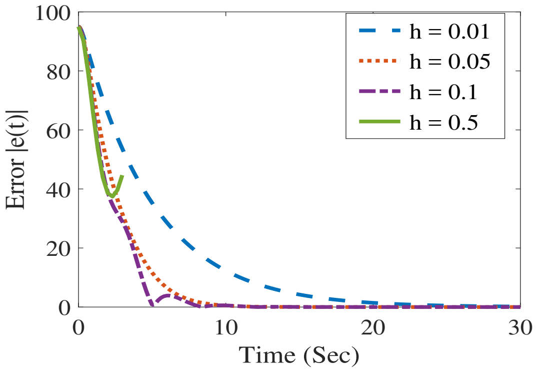

Figure 3: Error time traces for different values of time-delay

Similar to multiple values of delay, the effect of multiple gains can also be studied. The simulation results for this case are plotted in Figure 4. It can be seen from Figure 4 that as gain increases the system's transient response becomes slower. This is due to the time-delay in the system for which the high-gain can deteriorate the transient response. Therefore, the selection of gain through the relation ($\alpha = 1 + \frac{k_a}{h}$), ensures a gain and time-delay parameter pair, which can stabilize the system.

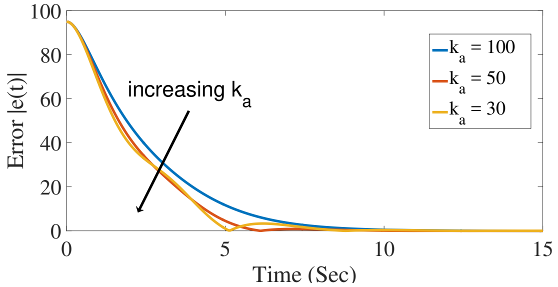

Figure 4: Error time traces for different values of gain

4 CONCLUSION

It has been noted that acceleration command for a visual servoing can results in smoother trajectory. However, such design based on Lyapunov requires derivative of the feature error, which is a state to the system but not measurable. In this work, the error derivative is approximated with a time-delay and stability analysis is carried out for the time-delayed approximated system to obtain the controller gains and bound on time-delay for stability. With the information of past data, the controller can make the camera track the desired object. Further, the effect of gain and time-delay is studied with the help of multiple simulations, finding that high delay can destabilize the system and high gain makes the performance sluggish.

5 FUTURE WORK

In this paper, the proposed controller works for systems with perfectly known camera parameters and pixel depth values. But to make it robust against approximation errors, a new approach can be developed in the future. Uncertainty in camera parameters and depth values will affect in calculation of the pseudo-inverse of the interaction matrix J_a. These uncertainties can be due to vision sensor noise and/or camera calibration errors which influence the controller performance [19]. The error in depth information is also responsible for the estimation of interaction matrix approximation [20]. Moreover, It is known that time delay method is not robust to error approximation [6]. So the modified error dynamics can be represented as

$$\ddot{e} = -\left(\frac{k_a}{h} + 1\right) e + \frac{k_a}{h} e_h + \delta \tag{52}$$

where δ accounts for uncertainties in pseudo-inverse of the interaction matrix J_a and error due to time-delayed approximation. So a robust analysis and a modified controller design are of utmost importance for future works in this domain.

REFERENCES

[1] F. Chaumette and S. Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics Automation Magazine*, 13(4):82–90, Dec 2006.

[2] AH Aabdul Hafez, Enric Cervera, and CV Jawahar. Hybrid visual servoing by boosting ibvs and pbvs. In *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, pages 1–6. IEEE, 2008.

[3] Peter I Corke and Seth A Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4):507–515, 2001.

[4] M. Keshmiri, W. Xie, and A. Mohebbi. Augmented image-based visual servoing of a manipulator using acceleration command. *IEEE Transactions on Industrial Electronics*, 61(10):5444–5452, Oct 2014.

[5] Joyjit Mukherjee, Spandan Roy, Indra Narayan Kar, and Sudipto Mukherjee. A double-layered artificial delay-based approach for maneuvering control of planar snake robots. *Journal of Dynamic Systems, Measurement, and Control*, 141(4):041012, 2019.

[6] S. Roy, S. Nandy, R. Ray, and S. N. Shome. Time delay sliding mode control of nonholonomic wheeled mobile robot: Experimental validation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2886–2892, May 2014.

[7] S. Roy, A. Patel, and I. N. Kar. Analysis and design of a wide-area damping controller for inter-area oscillation with artificially induced time delay. *IEEE Transactions on Smart Grid*, pages 1–1, 2018.

[8] A Galip Ulsoy. Time-delayed control of siso systems for improved stability margins. *Journal of Dynamic Systems, Measurement, and Control*, 137(4):041014, 2015.

[9] David A Forsyth and Jean Ponce. Computer vision: A modern approach. 2002.

[10] HA Martins, John R Birk, and Robert B Kelley. Camera models based on data from two calibration planes. *Computer Graphics and Image Processing*, 17(2):173–180, 1981.

[11] Joaquim Salvi, Xavier ArmanguÃĬ, and Joan Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7):1617 – 1635, 2002.

[12] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 1996.

[13] Mark W Spong, Seth Hutchinson, Mathukumalli Vidyasagar, et al. *Robot modeling and control*. 2006.

[14] Peter Corke. *Robotics,Vision and Control: Fundamental Algorithms In MATLAB*, volume 118. Springer-verleg, Australia, 2011.

[15] M. Keshmiri and W. F. Xie. Augmented imaged based visual servoing controller for a 6 dof manipulator using acceleration command. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 556–561, Dec 2012.

[16] Jean-Thierry Lapresté, Frédéric Jurie, Michel Dhome, and François Chaumette. An efficient method to compute the inverse jacobian matrix in visual servoing. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 1, pages 727–732. IEEE, 2004.

[17] Jack K Hale. Functional differential equations. In *Analytic theory of differential equations*, pages 9–22. Springer, 1971.

[18] Spandan Roy and Indra Narayan Kar. Adaptive robust tracking control of a class of nonlinear systems with input delay. *Nonlinear Dynamics*, 85(2):1127–1139, Jul 2016.

[19] Bernard Espiau. Effect of camera calibration errors on visual servoing in robotics. In *Experimental robotics III*, pages 182–192. Springer, 1994.

[20] Ezio Malis and Patrick Rives. Robustness of image-based visual servoing with respect to depth distribution errors. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 1, pages 1056–1061. IEEE, 2003.