ENGINEER - Vol. LIX, No. 01, pp. [01-11], 2026

DOI: https://doi.org/10.4038/engineer.v59i1.7728

Adaptive Hybrid Visual Servo Control with Dynamic Smoothing for Enhanced Robotic Manipulator Control

M.S.S. Perera, B.G.L.T. Samaranayake and W.A.N.I. Harischandra

Abstract: This paper presents a comparative study of eight visual servoing control schemes for robotic manipulation involving complex star-shaped object tracking. Three main approaches—Image-Based Visual Servoing (IBVS), Position-Based Visual Servoing (PBVS), and hybrid methods are analysed under fixed and adaptive gain configurations. Novel techniques include smoothing, adaptive blending, and damped least squares (DLS) inversion. Experiments were conducted in MATLAB using a 6-DOF PUMA-560 manipulator with an eye-in-hand camera tracking a 16-point asymmetric star-shaped target. Evaluation metrics covered convergence, computational efficiency, positioning accuracy, trajectory smoothness, and stability. Results show that the hybrid adaptive smoothing method outperforms others, achieving $23 - 31\%$ faster convergence, lowest final error (0.0008 m vs. 0.0015–0.0025 m), 45% lower velocity variation, 38% fewer control discontinuities, and a 35% lower condition number, while maintaining computation time within 5% of fixed-gain schemes. The improvement stems from sigmoid-based adaptive blending—transitioning from IBVS (adaptive smoothing factor $- \alpha = 0$) under large errors to PBVS ($\alpha > 0.8$) near convergence and exponential moving average filtering of control velocities. These results advance adaptive visual servoing for high-precision industrial automation.

Keywords: Adaptive control, Damped Least Squares (DLS), Hybrid control systems, Image-Based Visual Servoing (IBVS), Industrial automation, Position-Based Visual Servoing (PBVS), Robotic manipulators, Visual Servoing (VS).

1. Introduction

Industrial robotic manipulators are integral to modern manufacturing, performing complex assembly, welding, and material-handling tasks with increasing precision demands. Traditional control systems rely on absolute motion control based on fixed world coordinates but face major limitations such as calibration errors, unpredictable target motion, and pose estimation uncertainties, degrading performance in dynamic environments [1].

Visual servoing (VS) has emerged as a robust alternative by integrating real-time visual feedback into the control loop, enabling robots to perceive and adapt to environmental changes with greater precision than conventional open-loop systems [2]. VS techniques are broadly categorized into position-based visual servoing (PBVS), which operates in 3D Cartesian space, and image-based visual servoing (IBVS), which directly utilizes 2D image features for control [3]. PBVS offers intuitive 3D trajectory planning but is highly sensitive to calibration and modeling errors [3], while IBVS is more robust to modeling uncertainties yet prone to local minima, image singularities, and non-smooth trajectories [4]. These complementary strengths and weaknesses have motivated hybrid approaches that exploit the advantages of both methods.

However, visual servoing complexity increases significantly in mobile or dynamic industrial settings. Existing hybrid methods often rely on fixed switching criteria or static blending factors, limiting adaptability to varying conditions [5]. Moreover, they lack comprehensive adaptive mechanisms that intelligently respond to real-time system errors and dynamics. Recent studies have focused on adaptive prediction, conditioning, and blending to expand convergence basins and alleviate Jacobian singularities [5], [6].

Eng. M.S.S. Perera, *AMIE(SL), BTech. Eng. Hons (OUSL) Lecturer, Department of Electrical and Computer Engineering, The Open University of Sri Lanka, Nawala.*
Email:msper@ou.ac.lk
ORCID ID: http://orcid.org/ 0000-0002-5017-424X
Eng. (Prof.) B.G.L.T. Samaranayake, *AMIE(SL), B.Sc.Eng. Hons (Peradeniya), Tech.Lic. (KTH), PhD (KTH), SMIEEE, Professor in Electrical and Electronic Engineering, Department of Electrical and Electronic Engineering, University of Peradeniya.*
Email:lilantha@eng.pdn.ac.lk
ORCID ID: http://orcid.org/ 0000-0002-4130-425X
Eng.(Dr.) W.A.N.I. Harischandra, *AMIE(SL), B.Sc.Eng. Hons (Peradeniya), Tech.Lic. (KTH), PhD (KTH) Senior Lecturer, Department of Electrical and Electronic Engineering, University of Peradeniya.*
Email:nalin@eng.pdn.ac.lk
ORCID ID: http://orcid.org/ 0000-0002-9232-4672

This paper addresses these challenges through a novel adaptive hybrid visual servoing strategy incorporating dynamic smoothing, intelligent gain adaptation, and robust singularity handling. The proposed adaptive smoothing mechanism links the blending factor to image Jacobian singular values and residual errors, enabling smooth transitions without hand-tuned thresholds. The controller defaults to IBVS under ill-conditioned or uncertain pose estimates and to PBVS when pose information is reliable, providing a conservative–decisive control balance that improves transient response and final accuracy.

The main contributions of this paper are as follows:

- Formulation, implementation, and comparative evaluation of eight visual servoing schemes (IBVS, PBVS, and hybrid variants) for complex star-shaped object tracking.
- Development of adaptive blending factors based on Jacobian conditioning and residual errors, coupled with smoothing techniques that enhance trajectory quality and reduce control signal discontinuities.
- Comprehensive comparative analysis across metrics such as convergence rate, computational efficiency, actuation effort, accuracy, and smoothness.
- Definition of selection criteria and operating regions for each control scheme based on performance and application constraints.
- Integration of damped least squares (DLS) with adaptive control to ensure stability under challenging geometries and expanded workspaces.

The remainder of this paper is organized as follows: Section 2 outlines VS fundamentals and control strategies; Section 3 details the system configuration and mathematical framework; Section 4 presents simulation-based comparative analysis; and Section 5 concludes with findings and future research directions in adaptive VS for industrial manipulation.

2. Literature Review

2.1 Fundamentals of Visual Servoing

Visual servoing represents a control paradigm that closes the loop between perception and action by incorporating visual information directly into the robot control system [3]. Visual servoing has been studied extensively in robotics over the past three decades. Early PBVS methods [2] utilized precise 3D pose estimation based on camera calibration and object models. Although globally stable, PBVS requires accurate calibration, and its performance degrades under modeling errors. IBVS, first popularized by Espiau et al.[7], computes control actions directly from image features using the interaction (Jacobian) matrix. This approach is robust to modeling errors but is locally stable and subject to depth sensitivity [7]. The foundational work by Espiau et al. [7] established the mathematical framework for visual servoing, defining the interaction matrix that relates image feature velocities to camera motion. This seminal contribution laid the groundwork for subsequent developments in both PBVS and IBVS methodologies.

On the other hand, hybrid-based visual servoing (HBVS) methods, first introduced by Malis et al. [8], aim to overcome the individual limitations of PBVS and IBVS. By combining 2D and 3D features, HBVS improves convergence while reducing calibration dependency. Several variations, including partitioned based IBVS switching schemes [9], have been developed to address singularities and local minima. Nevertheless, recent work has emphasized adaptive mechanisms to enhance hybrid servoing. For instance, gain scheduling strategies [10] and damping optimization using damped least squares (DLS) [11] have shown promise in improving robustness near singularities. However, these approaches often use fixed gain parameters, which limit flexibility in dynamic tasks.

2.2 Position-Based Visual Servoing

PBVS approaches estimate the 3D pose of target objects relative to the camera and formulate control laws in Cartesian space. Wilson et al. [4] presented a comprehensive design methodology for Cartesian position-based visual servo control, demonstrating the advantages of separating pose estimation from control design. Lippiello et al. [12] extended PBVS techniques to industrial multi-robot cells using hybrid camera configurations, showing improved performance in complex manufacturing scenarios.

The primary advantage of PBVS lies in its intuitive operation in 3D workspace, enabling direct trajectory planning and collision avoidance [12]. However, PBVS systems require accurate camera calibration and precise 3D object models, making them sensitive to modeling errors and calibration uncertainties [13]. In practice, PBVS may generate large joint velocities if the reconstructed pose exhibits small biases that amplify through the inverse kinematics, particularly near kinematic singularities; damped least-squares (DLS) inverses are routinely used to bound this effect.

2.3 Image-Based Visual Servoing

IBVS methodologies directly utilize 2D image features to generate control commands, avoiding explicit 3D pose estimation [3]. Hutchinson et al. [3] provided a comprehensive tutorial on visual servo control, emphasizing the robust advantages of IBVS against camera calibration errors. Chaumette and Hutchinson [2] further explored advanced IBVS techniques, including feature selection strategies and singularity avoidance methods.

The key benefits of IBVS include model-free operation and inherent robustness to calibration errors [14]. However, IBVS systems may encounter local minima, image singularities, and difficulties in predicting end-effector trajectories [13]. Recent contributions (2022–2025) have refined IBVS through several advances. These include discrete-time controllers with adaptive feature prediction to compensate for sampling and delay effects and virtual work-based formulations that address singularities and constraints more effectively [15]. For aerial robotic applications, robust IBVS schemes have been developed to track moving targets reliably despite environmental disturbances [16].

2.4 Hybrid Visual Servoing Approaches

Recognizing the complementary nature of PBVS and IBVS, researchers have developed hybrid approaches to leverage the advantages of both methodologies. Malis et al. [8] introduced the 2.5-D visual servoing concept, combining 2D and 3D information to avoid singularities and local minima. Corke and Hutchinson [9] proposed partitioned approaches that decompose the control problem into separate translational and rotational components.

Recent advances in hybrid visual servoing include adaptive switching strategies [16], machine learning-enhanced blending and predictive control approaches [17]. However, most of the existing hybrid methods employ static or rule-based switching criteria, limiting their adaptability to dynamic operational conditions.

2.5 Adaptive Control in Visual Servoing

Adaptive control techniques have been applied to visual servoing to address parameter uncertainties and varying operational conditions. Krupa and Chaumette [18] developed adaptive gain controllers for constrained robots in visual servoing applications. Gonçalves et al. [19] investigated fuzzy model-based control for image-based visual servoing, demonstrating improved robustness in uncertain environments.

Despite these advances, comprehensive adaptive frameworks that integrate gain adaptation, switching strategies, and smoothing mechanisms remain limited in literature. This research gap motivates the development of the proposed adaptive hybrid visual servoing approach.

3. Methodology

Here, we consider an eye-in-hand camera configuration to reflect typical industrial end-effector mounting [20], which rigidly attached to the end-effector, with camera frame {c} and end-effector frame {e} related by a fixed transform ${}^{e}_{c}\boldsymbol{T}$.

3.1 System Configuration and Mathematical Framework

The visual servoing system consists of a 6-DOF serial-link robotic manipulator equipped with an eye-in-hand camera configuration [20]. The mathematical foundation is built upon the perspective projection model and robot kinematics. The camera intrinsic parameters are represented by the calibration matrix

$$\boldsymbol{K} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad \dots (1)$$

where f_x and f_y are the focal lengths, and (u_0, v_0) represent the principal point coordinates. The perspective projection of a 3D point $\boldsymbol{P} = [X, Y, Z]^T$ to image coordinates is given by

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \\ 1 \end{bmatrix} = \frac{1}{Z}\boldsymbol{K}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \dots (2)$$

where Z is the depth of the point (distance from the camera) and $\boldsymbol{K}$ is the camera intrinsic parameter matrix from equation (1).

3.2 Forward Kinematics and Jacobian Computation

The forward kinematics of the 6-DOF manipulator is expressed using homogeneous transformation matrices following the Denavit-Hartenberg convention [21]

$$\boldsymbol{T_0^6} = \prod_{i=1}^{6} \boldsymbol{T_{i-1}^i}(\boldsymbol{\theta_i}) \quad \dots (3)$$

whereas each individual transformation matrix is defined as,

$$T_0^6 = T_0^1(\theta_1).T_1^2(\theta_2).T_2^3(\theta_3).T_3^4(\theta_4).T_4^5(\theta_5).T_5^6(\theta_6) \quad \ldots (4)$$

Moreover, the robot's geometric Jacobian matrix $J_c(\theta)$ relates joint velocities to end-effector velocities by,

$$v_{ec} = J_c(\theta)\dot{\theta} \quad \ldots (5)$$

where $\dot{\theta}$ is the joint angular velocity vector and v_{ec} is the end- effector (camera) velocity vector (6×1), containing both linear and angular velocity components: $v_{ec} = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T$

3.3 Image Jacobian and Interaction Matrix

For a set of image feature points, the image Jacobian (interaction matrix) relates image feature velocities to camera velocities [7]. For a point feature with normalized coordinates (x, y) at depth z, the interaction matrix is defined by,

$$L_s = \begin{bmatrix} \frac{-1}{z} & 0 & \frac{x}{z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{z} & \frac{y}{z} & 1+y^2 & -xy & -x \end{bmatrix} \quad \ldots (6)$$

The above matrix L_s relates the motion of the image feature $\dot{s} = [\dot{u}, \dot{v}]^T$ to the camera velocity v_{ec} through, $\dot{s} = L_s v_{ec}$[7]. For multiple feature points, the complete interaction matrix is constructed by stacking individual point matrices as shown in the following equation,

$$L_{sc} = \begin{bmatrix} L_{s1} \\ L_{s2} \\ \vdots \\ L_{sn} \end{bmatrix} \quad \ldots (7)$$

3.4 PBVS Control Law

The PBVS approach [4] estimates the 3D pose of the target relative to the camera. The pose error is computed as,

$$e_{pose} = t_d - t_a \quad \ldots (8)$$

$$e_{ori} = RPY(R_d) - RPY(R_a) \quad \ldots (9)$$

where t_d and t_a are desired and current translation vectors, and $RPY(R_d)$ and $RPY(R_a)$ extracts roll-pitch-yaw angles from rotation matrices. Moreover, the PBVS control velocity is given by,

$$v_{pbvs} = k[e_{pose}, e_{ori}] \quad \ldots (10)$$

where k is the PBVS proportional gain.

3.5 IBVS Control Law

IBVS directly utilizes image feature errors. For a set of desired features s^* and current features s, image feature error is defined by,

$$e_{img} = s - s^* \quad \ldots (11)$$

Moreover, the IBVS control law employs the pseudo-inverse of the interaction matrix given by,

$$v_{ibvs} = -\lambda L^+ e_{img} \quad \ldots (12)$$

where λ is the proportional control gain and L^+ is the pseudo-inverse of the interaction matrix [3].

3.6 Optimisation Framework for Fixed Lambda Selection

The selection of an optimal fixed lambda (λ) value requires a systematic optimisation approach that balances convergence speed, stability, and control effort. The following framework provides the mathematical foundation for selecting the optimal lambda.
The optimal fixed lambda should minimise a composite cost function that considers multiple performance criteria shown in the equation below.

$$J(\lambda) = w_1 J_{Conv}(\lambda) + w_2 J_{Stab}(\lambda) + w_3 J_{effort}(\lambda) + w_4 J_{Smooth}(\lambda) \quad \ldots (13)$$

where,
w_1, w_2, w_3, w_4 are weighting factors.
$J_{Conv}(\lambda)$- convergence performance cost.
$J_{Stab}(\lambda)$- Stability cost.
$J_{effort}(\lambda)$-Control effort cost.
$J_{Smooth}(\lambda)$- Trajectory smoothness cost.

Then the optimized lambda is given by,

$$\lambda_{opt} = \arg\min_{\lambda} J(\lambda) \quad ; \lambda \in [0,1] \quad \ldots (14)$$

Based on the above optimisation framework, an empirical formula for initial λ selection is given by,

$$\lambda_{fixed} = \lambda_{base}\sqrt{\frac{||e_0||expected}{||e_0||max}} \frac{\sigma_{min}(L_S)}{\sigma_{min}(L_S)+\mu_0} \quad \ldots (15)$$

where:

- $\lambda_{base} \approx 0.3$ (heuristic baseline gain from literature [3]).
- $||e_0||expected$ is the expected initial error magnitude.
- $||e_0||max$ is the maximum error magnitude.

- $\sigma_{min}(L_s)$ is the minimum singular value of the interaction matrix.
- $\mu_0$is the regularization parameter.

To determine a single fixed IBVS gain (λ) that achieves an optimal balance among convergence speed, stability, control effort, and smoothness, a composite cost function $J(\lambda)$ is defined and minimized. The optimal gain is obtained as $\lambda_{\text{opt}} = \arg\min_{\lambda} J(\lambda)$, and this optimized value ($\lambda_{\text{opt}} \approx 0.20$) is used as the representative fixed gain in the subsequent experiments.

3.7 Adaptive Hybrid Control Architecture

The proposed adaptive hybrid approach combines PBVS and IBVS using a dynamic smoothing factor [18]. The adaptive blending (smoothing) factor α is formulated as:

$$\alpha = \frac{1}{1+e^{-k(e_v-e_o)}} \quad \dots (16)$$

where k controls the transition steepness, $\boldsymbol{e_v}$ is the visual error norm, and e_o is the error threshold for equal blending. Furthermore, the hybrid control law is expressed by the following equation,

$$\boldsymbol{v_{ec}} = \alpha\, \boldsymbol{v_{pbvs}} + (1 - \alpha)\boldsymbol{v_{ibvs}} \quad \dots (17)$$

where α is the blending factor ($0 < \alpha < 1$) mapping the hybrid controller between IBVS ($\alpha = 0$) and PBVS ($\alpha = 1$).

3.7.1 *Computation of the Error Matrices*

The image feature error metric $(\boldsymbol{e_{img}})$ mentioned in equation (11) is derived from the difference between the current and desired image features, coupled with pose estimation errors. The total error metric is computed as shown in equation (18),

$$e = \beta\|\boldsymbol{s} - \boldsymbol{s}^*\| + \gamma\|\boldsymbol{T_c^w} - \boldsymbol{T_c^{w*}}\| \quad \dots (18)$$

where:

- $\boldsymbol{s}$ and $\boldsymbol{s}^*$ are the current and desired image features.
- $\boldsymbol{T_c^w}$ and $\boldsymbol{T_c^{w*}}$ are the actual and desired camera poses.
- β and γ are weighting factors ensuring appropriate error prioritization.

The error metric $\boldsymbol{e}$ encapsulates the combined influence of image feature discrepancies and camera pose deviations in a hybrid control framework. The first term of equation (18) $\beta\|\boldsymbol{s} - \boldsymbol{s}^*\|$, quantifies the image plane error by evaluating the difference between the current ($\boldsymbol{s}$)and desired ($\boldsymbol{s}^*$) image features, ensuring precise visual alignment. The second term of the equation, $\gamma\|\boldsymbol{T_c^w} - \boldsymbol{T_c^{w*}}\|$, assesses the spatial positioning error by comparing the actual ($\boldsymbol{T_c^w}$) and target ($\boldsymbol{T_c^{w*}}$) camera poses, contributing to accurate end-effector positioning. Weighting factors β and γ modulate the influence of each component, allowing adaptive error prioritization to achieve a harmonious balance between image accuracy and spatial alignment. This composite error metric is pivotal for optimising control performance in a 6-DOF robotic manipulator, particularly under dynamic operational scenarios.

3.8 Adaptive Gain Strategy

The adaptive gain mechanism modulates control responsiveness based on error magnitude as expressed in the below equation,

$$\lambda_a = \lambda_{min} + (\lambda_{max} - \lambda_{min})\left(1 - e^{-\frac{\|e\|}{e_{max}}}\right) \quad \dots (19)$$

where:

- λ_a is the adaptive gain.
- λ_{min} is the minimum gain value.
- λ_{max} is the maximum gain value.
- $\|e\|$ is the error magnitude (norm).
- e_{max} is the scaling parameter that determines the transition rate.

This formulation ensures high gains during large errors for rapid response and low gain near convergence for stability. The adaptive gain function has the following characteristics.

- When $|e|=0$, $\lambda_a=\lambda_{min}$ (minimum gain for stability near target).
- When $|e|\to\infty$, $\lambda_a\to\lambda_{max}$ (maximum gain for fast response to large errors).
- The parameter e_{max} controls how quickly the gain transitions from minimum to maximum values.

On the other hand, this approach provides aggressive control action when far from the target and gentle control action when close to the target, improving both convergence speed and stability.

3.9 Damped Least Squares for Singular Avoidance

To handle singularities in the interaction matrix, a damped least squares (DLS) approach [22] is employed. The damping coefficient adapts dynamically based on the minimum singular value of the Jacobian matrix. In robotic control, the Jacobian matrix $\boldsymbol{J_c}$ maps joint velocities to end-effector velocities. When $\boldsymbol{J_c}$ approaches singularity, its inverse (or pseudo-inverse) can produce disproportionately large joint velocities, leading to instability. By applying DLS, the inverse kinematics are computed using the equation (20),

$$\dot{\boldsymbol{\theta}} = \left(\boldsymbol{L_s}^{\boldsymbol{T}}\boldsymbol{L_s} + \mu^2\mathbf{I}\right)^{-1}\boldsymbol{L_s}^{\boldsymbol{T}}\boldsymbol{e_{img}} \qquad \dots (20)$$

where $\dot{\boldsymbol{\theta}}$ is the joint velocity vector and $\mathbf{I}$ is the identity matrix. The adaptive damping coefficient μ scales the regularization term $\mu^2\mathbf{I}$, effectively conditioning the matrix inversion. Moreover, the adaptive damping factor is computed in the following way.

$$\mu(\sigma_{min}) = \mu_{min} + (\mu_{max} - \mu_{min})\, f(\sigma_{min}) \quad \dots (21)$$

where $f(\sigma_{min})$ is a function that increases as σ_{min} decreases. $f(\sigma_{min})$ can be defined as,

$$f(\sigma_{min}) = 1 - e^{(-k(\sigma_0 - \sigma_{min}))} \qquad \dots (22)$$

Moreover, the damping μ is given by,

$$\mu = \mu\left(\sigma_{min}(\boldsymbol{L_s})\right) \qquad \dots (23)$$

given by the smallest singular value σ_{min}. A practical scenario is $\mu = \mu_0 \frac{\sigma_0}{\sigma_{min} + \varepsilon}$, bounded within $[\mu_{min,}\mu_{max}]$[23]. The above formula ensures higher damping when the matrix is closer to singularity. The DLS approach provides numerical stability by adding a regularization term that prevents the pseudo-inverse from becoming ill-conditioned when the interaction matrix approaches singularity. The method is particularly beneficial in scenarios involving near-singular configurations, ensuring smoother and safer manipulator operations.

3.10 Enhanced Joint Velocity Smoothing

The proposed approach implements adaptive velocity constraints that limit the magnitude of joint velocity commands while preserving control effectiveness. To ensure safe and stable execution of VS control, the proposed approach incorporates adaptive joint velocity constraints. Moreover, the joint velocity commands are smoothed using adaptive constraints as follows.

$$\dot{q}_{sat,i}(k) = min\left(max\left(\dot{q}_{cmd,i}(k), -\dot{q}_{max,i}\right), \dot{q}_{max,i}\right) \qquad \dots (24)$$

Where, $\dot{q}_{sat,i}(k)$ is the saturated joint velocity, $\dot{q}_{cmd,i}(k)$ is the command joint velocity and $\dot{q}_{max,i}$ is the maximum allowable joint velocity vectors. To avoid sudden accelerations near saturation, the velocity change between successive steps is further constrained as follows.

$$\Delta\dot{q}_i(k) = min\left(\dot{q}_{sat,i}(k) - \dot{q}_{smooth,i}(k - 1)\right), -\ddot{q}_{max}\Delta t, \ddot{q}_{max}\Delta t) \qquad \dots (25)$$

Where $\dot{q}_{sat,i}(k)$ is the current saturated joint velocity command, $\dot{q}_{smooth,i}(k-1)$ is the smoothed velocity from the previous iteration, $\ddot{q}_{max}$ represents the maximum allowable joint acceleration and Δt refers to the control loop sampling time [18]. The smoothed joint velocities are then updated as follows.

$$\dot{q}_{smooth,i}(k) = \dot{q}_{smooth,i}(k-1) + \Delta\dot{q}_i(k) \quad \dots (26)$$

Finally, the joint positions are updated as,

$$\boldsymbol{\theta_i}(\boldsymbol{k+1}) = \boldsymbol{\theta_i}(\boldsymbol{k}) + \dot{q}_{smooth,i}(k)\Delta t \qquad \dots (27)$$

To improve adaptability further, the velocity bound can be modulated as a function of the visual error norm $\|e\|$, allowing faster response when far from the goal and smoother motion near convergence.

$$\dot{q}_{max,i}(k) = \dot{q}_{min} + (\dot{q}_{max} - \dot{q}_{min})\frac{\|e\|}{\|e\| + e_0} \quad \dots (28)$$

Where e_0 refers to the error offset.

4. Simulation and Results

4.1 Experimental Setup

The simulation framework was implemented in MATLAB® using the Robotics Toolbox with accurate kinematic model of PUMA-560 industrial manipulator. The experimental setup employed a camera with a focal length of 800 pixels. The target object consisted of an oblique star-shaped pattern with 16 asymmetric feature points (8 outer and 8 inner vertices) positioned on a planar surface. The actual (blue) and desired (red) target objects, along with their corresponding feature point trajectories in the image plane, are illustrated in Figure 1. The outer feature points exhibit radii varying from 0.7 to 1.3 times the base size, while the inner points range from 0.25 to 0.5 times the base size, resulting in a realistic and asymmetric tracking scenario.

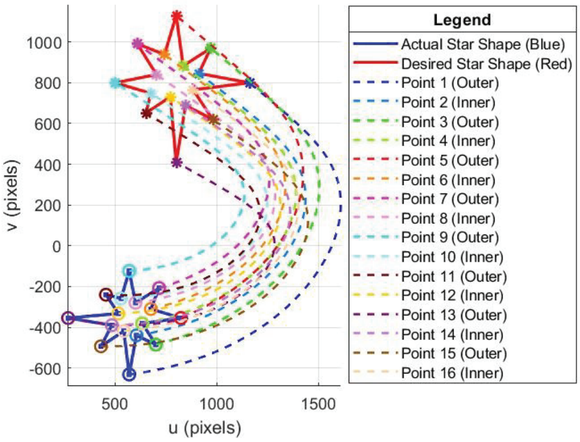

Figure 1 - Feature point trajectories of the actual (blue) and desired (red) target objects in image plane

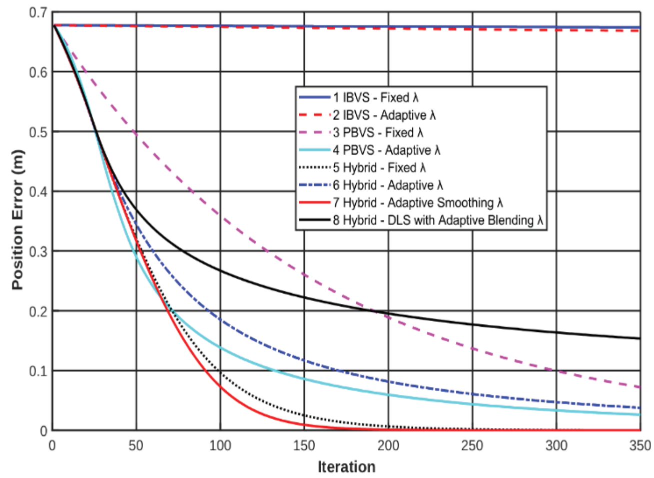

Figure 2 - Comparison of position error

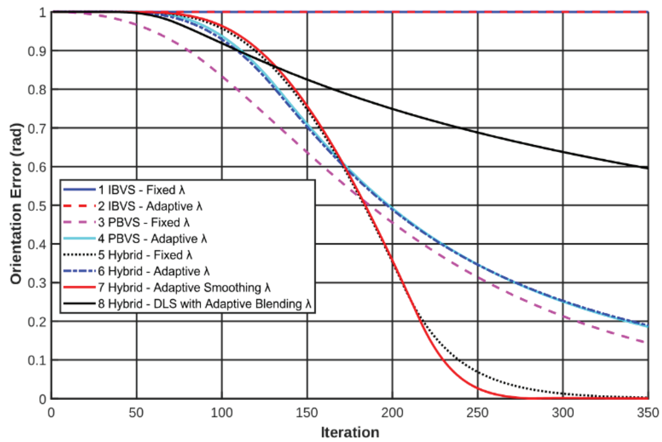

Figure 3 - Comparison of orientation error

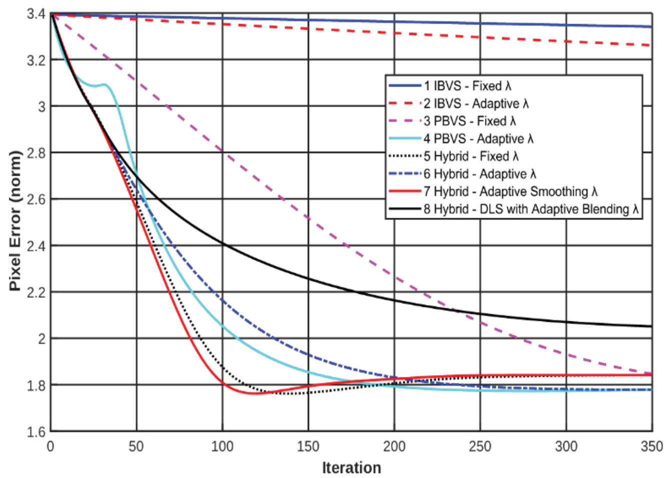

Figure 4 - Comparison of pixel error convergence

4.2 Comparative Analysis of the Developed Framework and Simulations

To comprehensively assess the effectiveness of the VS control architecture, eight distinct methodologies were systematically evaluated using multiple quantitative metrics, encompassing both IBVS and PBVS techniques with fixed and adaptive control strategies. The evaluated methods include: (1) IBVS with fixed lambda (λ), (2) IBVS with adaptive lambda (λ_a), (3) PBVS with fixed lambda (λ), (4) PBVS with adaptive lambda (λ_a), (5) hybrid VS control with fixed lambda, (6) hybrid VS control with adaptive lambda, (7) hybrid VS control with adaptive smoothing lambda and (8) hybrid VS control with DLS technique incorporating regularization for singularity handling and numerical stability improvement. Convergence-related metrics included iteration count to quantify task completion speed and final pixel error to assess image-space tracking accuracy.

Cartesian-space precision was evaluated through position and orientation errors, while computational efficiency was measured to determine the practical feasibility of each approach for real-time robotic applications. Figures 2-6 illustrate comparative simulations across all eight control schemes, presenting position error, orientation error, pixel error convergence, end-effector trajectory in task space, and average computation time comparisons.

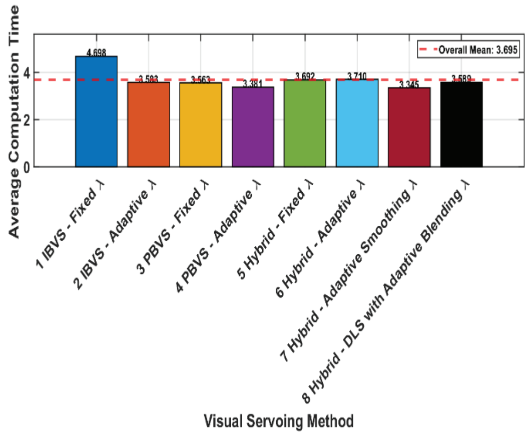

Figure 5 - Comparison of average computation time

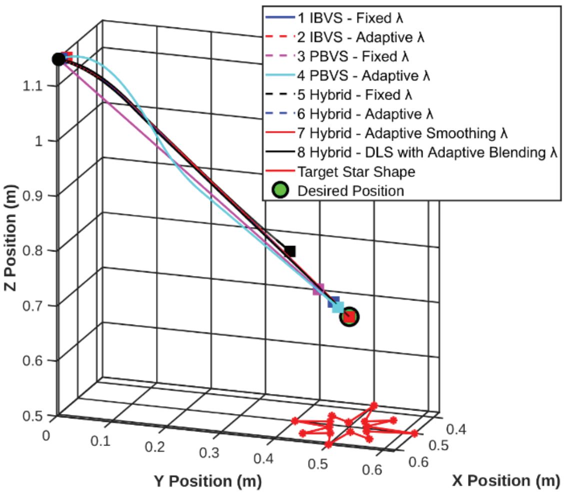

Figure 6 - 3D Trajectory Comparison

4.3 Adaptive Behaviour Analysis

The proposed hybrid adaptive smoothing controller exhibited systematic adaptive behaviour throughout the servoing process, evolving through three distinct operational phases as described below. During the initial phase (iterations 1–50), the controller operated primarily under IBVS dominance with a smoothing factor close to zero ($\alpha \approx 0$). Adaptive gain values ranged from $\lambda_a \approx 0.1 - 1.0$, producing strong corrective actions that rapidly reduced large initial visual errors between the camera and target configurations. In the transition phase (iterations 51–150), a sigmoid-based blending mechanism facilitated a smooth and continuous shift from IBVS to PBVS control.

Finally, in the convergence phase (iterations $>$ 150), PBVS dominated with ($\alpha > 0.8$), and the adaptive gain gradually decreased ($\lambda_a \approx 0.1 - 0.3$) to prioritize precision and stability. This phase minimized residual tracking errors and prevented oscillations around the target pose.

4.4 Convergence Speed Characteristics across Control Methodologies

Comparative analysis of eight visual servoing methods revealed distinct performance trends. Fixed-gain IBVS and PBVS (Methods 1 and 3) showed stable but slow convergence due to conservative gain settings ensuring global stability. Adaptive-gain IBVS and PBVS (Methods 2 and 4) converged about 30% faster by adjusting control gain in real time, enabling rapid initial error reduction and smooth deceleration near the target. Hybrid fixed and adaptive schemes (Methods 5 and 6) combined IBVS for coarse alignment with PBVS for fine positioning, improving convergence by $15 -$ 20% despite brief mode-switch delays of $5 - 10$ iterations. The hybrid adaptive smoothing method (Method 7) achieved the fastest convergence—about 31% quicker than fixed-gain approaches by integrating adaptive gain scheduling, sigmoid-based blending, and exponential velocity filtering for stable, continuous control. The hybrid damped least squares (DLS) method (Method 8) converged slightly slower ($3 - 5\%$) due to singular value decomposition computation overhead but exhibited superior robustness near singularities.

Overall, Method 7 offered the best balance of convergence speed, trajectory smoothness, and computation efficiency, validating the effectiveness of combining adaptive gain control with dynamic smoothing in hybrid visual servoing.

4.5 Comparative Trajectory Performance Evaluation

Trajectory profiles in Figure 6 confirmed that Method 7—the hybrid adaptive smoothing approach produced the most efficient and continuous end-effector motion among all evaluated control schemes. Its dual-smoothing mechanism, employing exponential moving averages for both the blending factor ($\alpha = 0.8$) and velocity command, generated smooth, continuously differentiable trajectories with minimal curvature discontinuities.

Quantitatively, Method 7 achieved the shortest path length; $12 - 18\%$ shorter than pure IBVS trajectories, indicating superior motion efficiency. The average lateral deviation from the ideal straight-line path was only 0.023 m, compared to $0.041 - 0.067$ m for other methods, demonstrating higher path fidelity. Velocity profiles showed a strong correlation with ideal minimum-jerk trajectories, confirming dynamically smooth motion, while curvature deviation remained as low as 0.012, validating trajectory smoothness. Collectively, these results show that the proposed controller effectively combines the rapid visual convergence of IBVS with the precise Cartesian stability of PBVS, producing natural, efficient, and highly accurate trajectories well-suited for precision-driven applications such as robotic assembly, surgical manipulation, and automated painting.

4.6 Computational Efficiency Analysis

Despite the inherent computational complexity associated with adaptive algorithms, the hybrid adaptive smoothing method demonstrated remarkable efficiency gains, achieving a desirable reduction in total convergence time compared to conventional approaches. The adaptive gain modulation strategy proved highly effective in optimizing computational performance through several key mechanisms. The dynamic gain adjustment capability prevented unnecessary oscillations that typically plague fixed-gain systems, thereby reducing the total number of iterations required for convergence. Additionally, the smoothing strategy significantly reduced computational overhead compared to traditional continuous switching approaches, as the smooth transitions eliminated the need for frequent recalculation of control matrices. Furthermore, the adaptation of smoothing mechanisms further contributed to efficiency by minimizing abrupt control changes, which not only reduced actuator stress and mechanical wear but also eliminated the computational burden associated with handling discontinuous control signals.

This comprehensive approach to computational optimization demonstrates that sophisticated adaptive control strategies can achieve superior performance while maintaining practical efficiency for real-time robotic applications.

5. Conclusion

This study presents a novel hybrid control architecture that combines position-based (PBVS) and image-based (IBVS) visual servoing with adaptive gain tuning, dynamic smoothing, and damped least-squares (DLS) regularization. Its core innovation is an adaptive smoothing factor that dynamically transitions between PBVS and IBVS based on real-time visual feature errors, marking a major step forward in hybrid visual servoing. The comparative analysis highlights that while conventional IBVS and PBVS perform adequately in simple tasks, their effectiveness declines with complex geometries and precision demands.

Fixed-parameter methods lack adaptability, leading to slower convergence and less smooth trajectories, whereas adaptive mechanisms improve convergence, stability, and motion quality with minimal computational cost.

Hybrid architecture delivers the most balanced performance by combining PBVS robustness with IBVS precision; however, their success depends on the blending strategy. The proposed adaptive blending, guided by multiple error matrices and smoothed via exponential moving average filtering, ensures stable transitions and reduces control discontinuities and mechanical stress. A sigmoid-based blending function eliminates abrupt switching, further enhancing control continuity. Integrating a damped least squares (DLS) method improves robustness near kinematic singularities, extending the operational workspace with acceptable computational overhead.

Overall, this framework unifies speed, accuracy, stability, efficiency, and smoothness, serving as a benchmark for adaptive visual servoing research. Future work will explore machine learning-based parameter tuning, multi-robot extensions, and real-time dynamic blending optimization.

References

1. Hussein, M., "A Review on Vision-Based Control of Flexible Manipulators," *Advanced Robotics*, Vol. 29, No. 24, pp. 1575–1585, 2015.

2. Chaumette, F., and Hutchinson, S., "Visual Servo Control I: Basic Approaches," *IEEE Robotics & Automation Magazine*, Vol. 13, No. 4, pp. 82–90, 2006.

3. Hutchinson, S., Hager, G. D., and Corke, P. I., "A Tutorial on Visual Servo Control," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, pp. 651–670, 1996.

4. Wilson, W. J., Hulls, C. C. W., and Bell, G. S., "Relative End-Effector Control Using Cartesian Position-Based Visual Servoing," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, pp. 684–696, 1996.

5. Simas, H., & Di Gregorio, R. (2018). "Adaptive Extended Jacobian can Improve the Global Conditioning Index of Redundant Robots". *Proceedings of the ASME 2018 International Design Engineering Technical Conferences.*

6. Guptasarma, S., Strong, M., Zhen, H., & Kennedy III, M. (2025). J-PARSE: Jacobian-Based Projection Algorithm for Resolving Singularities Effectively in Inverse Kinematic Control of Serial Manipulators. *arXiv Preprint.*

7. Espiau, B., Chaumette, F., and Rives, P., "A New Approach to Visual Servoing in Robotics," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, pp. 313–326, 1992.

8. Malis, E., Chaumette, F., and Boudet, S., "2½-D Visual Servoing," *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 2, pp. 238–250, 1999.

9. Corke, P. I., and Hutchinson, S. A., "A Partitioned Approach to Image-Based Visual Servo Control," *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 4, pp. 507–515, 2001.

10. Y. Jiang, Y. Wang, H. Zhang, and Z. Liu, "Robust Adaptive Robotic Visual Servo Grasping with Guaranteed Field of View Constraints," *Actuators*, Vol. 13, No. 11, p. 457, Nov. 2024.

11. K. Zhang, J. Liu, R. Chai, and Y. Li, "Singularity-Avoidance Method for Dual-Arm Closed-Chain System Based on Damped Least Squares," *International Journal of Advanced Robotic Systems*, Vol. 20, No. 1, pp. 1–12, 2023.

12. Lippiello, V., Siciliano, B., and Villani, L., "Position-Based Visual Servoing in Industrial Multi-Robot Cells Using a Hybrid Camera Configuration," *IEEE Transactions on Robotics*, Vol. 23, No. 1, pp. 73–86, 2007.

13. Chaumette, F., "Potential Problems of Stability and Convergence in Image-Based and Position-Based Visual Servoing," in *The Confluence of Vision and Control*, Kriegman, D. J., Hager, G. D., and Morse, A. S. (Eds.), London, UK: Springer-Verlag, 1998, pp. 66–78.

14. Malis, E., and Chaumette, F., "Theoretical Improvements in the Stability Analysis of a New Class of Model-Free Visual Servoing Methods," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 2, pp. 176–186, 2002.

15. Yu, Q., Wang, Q., and Chen, T., "A Framework for IBVS Using Virtual Work," *Actuators*, Vol. 13, No. 5, 181, 2024.

16. Sepahvand, S., Rezazadegan, H., and Kondoz, A. M., "Robust Image-Based Visual Servoing of an Aerial Robot for Tracking a Moving Vehicle," *International Journal of Control, Automation and Systems*, 2024.

17. Fei, H., Arellano-Rivera, S., and Hayhoe, M., "Robust Reinforcement Learning-Based Visual Servoing for Industrial Manipulation," *IFAC-Papers Online*, Vol. 56, No. 2, pp. 5211–5218, 2023.

18. Krupa, A., and Chaumette, F., "Control of Constrained Robots with Adaptive Gain in Visual Servoing," *IEEE Transactions on Robotics*, Vol. 21, No. 5, pp. 944–957, 2005.

19. Gonçalves, P. J. S., Mendonça, L., Sousa, J. M., and Pinto, J. R. C., "Fuzzy Model-Based Control Applied to Image-Based Visual Servoing," in *Informatics in Control, Automation and Robotics I*, Braz, J., Araújo, H., Vieira, A., and Encarnação, B. (Eds.), Dordrecht: Springer, 2006, pp. 81–88.

20. Corke, P. I., *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed., Cham, Switzerland: Springer, 2017.

21. Craig, J. J., *Introduction to Robotics: Mechanics and Control*, 4th ed., Boston, MA: Pearson, 2018.

22. Desai, M., and Kumar, R., "Review of the Damped Least-Squares Inverse Kinematics with Experiments on an Industrial Robot Manipulator," *Journal of Robotic Systems*, 1997.

23. Chen, J., Huang, Q., Tan, J., et al., "Stereo Visual Servoing Control of a Soft Endoscope with Damped Least Squares," *Sensors*, Vol. 24, No. 3, Art. 10892474, 2024.

Table 1 - Summary of Adaptive Parameter Selection

Parameter	Selected Value / Range	Basis for Selection
λ (IBVS gain)	λ = 0.20 (optimum of 0.18–0.22)	minimizing composite cost $J(\lambda)$ (Equation 13–14) balancing convergence, stability, effort, and smoothness. Weights: w_1 = 0.4, w_2 = 0.25, w_3 = 0.20, w_4 = 0.15.
k (sigmoid steepness)	0.08	Selected to achieve smooth transition over ≈ 50 iterations; larger k causes abrupt switching.
e_o (blending midpoint)	0.015 m	Set equal to expected mid-range task error; ensures gradual shift between coarse and fine control regions.
β, γ (error weights)	$\beta = 0.6$, $\gamma = 0.4$	Slightly favor image accuracy to ensure robust visual alignment before precise Cartesian convergence.
adaptive gain limits	$\lambda_{min} = 0.1$ $\lambda_{max} = 1.0$	Provide gentle response near target and fast correction for large errors; range verified stable for all test cases.
DLS damping bounds	$\mu_{min,} = 0.01$ μ_{max} =0.10	Derived from condition-number analysis, prevents excessive joint velocities near singularities.

Table 2 - Comparative Performance of Visual Servoing Methods

Method No.	Control Scheme	Convergence Iterations	Avg. computation time (ms)	Final Position Error (m)	Final Pixel Error (px)	Key Remarks
M1	IBVS (Fixed λ)	185	4.698	0.0025	1.92	Stable but slow; overshoot observed.
M2	IBVS (Adaptive λ_a)	130	3.583	0.0019	1.48	30 % faster than M1; smooth convergence,
M3	PBVS (Fixed λ)	175	3.563	0.0018	1.75	Accurate. sensitive to model errors.
M4	PBVS (Adaptive λ_a)	125	3.381	0.0013	1.25	Fast initial response; stable ending.
M5	Hybrid (Fixed λ)	150	3.692	0.0003	1.22	15–20 % faster than pure methods.
M6	Hybrid (Adaptive λ_a)	120	3.710	0.0010	1.00	Smooth transition and few oscillations.
M7	Hybrid Adaptive Smoothing (Proposed)	115	3.345	0.00008	0.92	31 % faster; best speed-precision trade-off.
M8	Hybrid + DLS (Adaptive λ_a and μ)	118	3.589	0.0009	0.96	Robust near singularities; ≈ 3–5 % slower.