Design and Experimental Validation of an IBVS-Controlled Mecanum Rover as a Testbed for Port-Hamiltonian Visual Servoing

K. Kiewiet[1], Prof. M. Muñoz Arias[2], Prof. S. Ahmed [3], Prof. M. Ito[4]

Abstract— **Deploying mobile robots in unstructured environments requires control strategies that go beyond classical kinematic approaches when vision is the primary feedback channel. Port-Hamiltonian (pH) visual servoing embeds damping and passivity guarantees directly into the control law, yet its predicted advantages over classical Image-Based Visual Servoing (IBVS) remain unverified on physical hardware. This thesis designs and validates a mecanum rover testbed for a rigorous comparison between the Sugiura–Hashimoto PI-IBVS and a reduced pH-IBVS controller. A visual interaction matrix is derived for bounding-box features, revealing a structural decoupling between lateral alignment and depth regulation enabled by the holonomic drive. An integrated three-layer architecture, comprising a YOLOv5 pipeline on a Hailo-8L accelerator at 30 Hz, an IBVS control layer, and a 50 Hz encoder-based inner wheel loop, provides all signals required for both controllers. A five-stage experimental campaign evaluates both laws on feature convergence, feature-space trajectory, commanded body velocities, and cumulative L_2 actuation cost. The pH controller achieves a 34.4% reduction in cumulative actuation cost and a 25.7% reduction in peak yaw rate compared to the SH-baseline, at the expense of a longer depth-channel settling time (11.9 s versus 7.1 s). These results constitute the first hardware-level evidence that even a velocity-level pH-IBVS reduction yields measurable energy-efficiency and transient-smoothness gains, showing strong potential for extension to full torque-level energy-based visual servoing on mobile platforms.**

I. Introduction

Modern manufacturing, logistics, agriculture, and service robotics increasingly operate in dynamic, unstructured environments where the foundational assumptions of traditional automation; fixed camera positions, pre-programmed trajectories, and fully calibrated workspaces, no longer hold. Applications ranging from flexible assembly and bin picking to agricultural harvesting and human-robot collaboration face challenges of high product variability, unknown object poses, changing illumination, occlusions, and moving targets [1], [2]. These scenarios require adaptive, real-time, vision-based closed-loop control that can accommodate uncertainty without scene recalibration.

A concrete example of these challenges arises in the Small Size League (SSL). The league's *Vision Blackout Challenge* requires each robot to replace the centralised overhead camera system with on-board perception, autonomously detecting and intercepting a moving ball under rapidly changing conditions [3], [4], [5]. This competition scenario mirrors the core industrial challenge: closing a control loop around visual measurements of a dynamic scene at high speed and low cost using only on-board sensors.

Such environments require sensors capable of providing high-rate feedback with minimal infrastructure. Recent breakthroughs in real-time object detection, particularly architectures such as YOLO [6], [7], have transformed the capabilities of commodity hardware: a standard webcam paired with frame-rate detection can now extract both angular position and an apparent-size depth proxy directly from bounding boxes, without geometric pre-calibration. This combination of accessible hardware and robust feature extraction enables vision to serve as the primary feedback channel for closed-loop control in unstructured settings [1], [8].

Formalising the camera as a feedback channel leads to the paradigm of visual servoing. There are two principal approaches. Position-Based Visual Servoing (PBVS) reconstructs a three-dimensional target pose from the image and is therefore sensitive to calibration errors and modeling inaccuracies. Image-Based Visual Servoing (IBVS), by contrast, operates entirely in the image plane: it drives a vector of image features $\mathbf{s}(t)$ toward a desired configuration $\mathbf{s}^*$ via the pseudo-inverse of the interaction matrix (image Jacobian), without requiring a three-dimensional model of the scene [1], [9], [10]. This intrinsic robustness to calibration drift is particularly valuable for mobile platforms, where vibrations or terrain irregularities continuously affect the camera-to-body geometry. Accordingly, IBVS has become the dominant visual-servoing strategy for wheeled robots that must cope with unstructured scenes, from differential-drive navigation [11] to omnidirectional platforms intercepting moving objects with on-board cameras alone [4], [5].

A practical limitation of the classicla IBVS law, however, is that a single scalar gain λ cannot independently shape the convergence of each feature channel: the horizontal alignment may have different sensitivities and ranges than the vertical-or depth alignment. The Sugiura-Hashimoto refinement of the classical IBVS law [5], [12] addresses this by replacing the scalar gain with a diagonal gain matrix $\mathbf{K}_P$ that assigns independent convergence rates per feature channel, and adds per-channel integral action that effectively rejects steady-state errors arising from motor dead zones, encoder quantisation, and modelling mismatch. Despite these improvements, both classical and PI-driven IBVS laws operate at the kinematic level: they prescribe velocity commands without accounting for inertia, internal dissipation, actuator limits, or dynamic coupling. As a result, fast maneuvers or external disturbances can excite dynamic modes that the

[1]Koen Kiewiet, [2]Prof. M. Muñoz Arias, [3] Prof. S. Ahmed and [4]Prof. M. Ito. [1,2,3] are with the Faculty of Science and Engineering, University of Groningen, Groningen, The Netherlands, and [4] is with Aichi Prefectural University, Japan. Corresponding author: K.kiewiet@student.rug.nl

kinematic controller cannot anticipate, degrading tracking performance or even destabilize the closed loop [13].

Addressing this limitation requires a control framework that accounts for the physical dynamics of the robot rather than ignoring them. The port-Hamiltonian (pH) formalism provides such a framework: it describes mechanical systems in terms of how they store energy, how they dissipate it, and how energy flows between subsystems [14], [15], [16]. Controllers designed within this framework, known as Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC) [17], reshape the total energy of the closed-loop system so that it reaches its minimum exactly at the desired target configuration, while additional damping ensures that the system settles there without oscillation.

Building on earlier applications of the pH framework to robotic manipulators with visual feedback [18], [19], Muñoz-Arias et al. [20] recently extended this approach to mobile robots with camera-based control. Their principal contribution is the treatment of gyroscopic and Coriolis forces: rather than cancelling these velocity-dependent terms as classical computed-torque methods do, they are explicitly incorporated into the damping injection step, preserving the energy-based stability guarantees of the pH structure. Additionally, integral action is introduced directly within the pH formulation through a leaky integrator that rejects persistent disturbances while providing structural anti-windup, a mechanism not present in the earlier manipulator-focused pH-IBVS works [19]. The closed-loop Hamiltonian, which serves as a Lyapunov function by construction in any pH system, can then be monitored in real time as a scalar certificate of controller performance. Simulations on a 6DOF-robotic manipulator show improved disturbance rejection and smoother velocity commands compared to kinematic IBVS [20]; yet no experimental implementation has verified these predictions. The Sagiuro-Hashimoto PI-IBVS has been validated on physical hardware in limited experiments [5], but the pH formulation remains untested outside simulation. In particular, no study has deployed pH-IBVS on a mobile rover with real-time vision and encoder feedback. This thesis adresses this gap with the following contributions:

1) **Interaction matrix for bounding-box features.** A visual Jacobian $\mathbf{J}_{\text{vis}} \in \mathbb{R}^{2\times 3}$ is derived via $\mathbf{J}_{\text{vis}} = \mathbf{J}_f \mathbf{J}_c$ for the reduced feature vector $\mathbf{s} = [\mu, \ell]^\top$, revealing the decoupling properties specific to the mecanum platform (Section II-D).
2) **Integrated experimental testbed.** A three-layer mecanum platform is presented that supplies all signals required for both PI-IBVS and pH-IBVS: real-time bounding-box features at 30 Hz, encoder-based wheel velocity feedback at 50 Hz, and an apparent-size depth proxy derived from the calibrated pinhole model (Sections IV–V).
3) **Experimental comparison on five performance metrics.** Closed-loop convergence of $\mu \to 0$ and $\ell \to \ell^*$ is demonstrated for both the SH PI-IBVS and a velocity-level pH-IBVS. The two controllers are compared on feature convergence, commanded body velocities, cumulative L_2 actuation cost $\mathcal{E}$, error norm $\|\mathbf{e}\|$, and feature-space trajectory, providing the first hardware-level evidence of the efficiency and transient-behaviour differences predicted by [20] (Sections VI–VII).
4) **Open-source implementation.** The complete control software, comprising the SH PI-IBVS and pH-IBVS control laws, the vision pipeline, the inner-loop wheel controller and plotting experiments, is released as a documented Python codebase to facilitate reproduction and extension of the reported experiments.[1]

An important remark is that the current platform commands body velocities rather than motor torques, so the pH law is implemented in its reduced form with ($\mathbf{M} = \mathbf{I}$) (Section VII-C). This reduction preserves the damping-injection and leaky-integral mechanisms that distinguish the pH controller from the SH-baseline, but it removes the inertia-shaping and Coriolis-embedding terms that provide the full energy-based stability guarantees of [20]. The experimental comparison therefore evaluates the practical kinematic-level benefits of the pH structure: smoother transient and reduced actuation cost through damping injection, while full torque-level deployment with identified motor dynamics is left as future work.

The remainder of this paper is organised as follows. Section II derives the IBVS framework and the visual interaction matrix. Section III presents the PI-IBVS and pH-IBVS control laws. Section IV describes the hierarchical software architecture. Section V details the hardware, encoder characterisation, and calibration. Section VI defines the experimental protocol, and Section VII reports the results and controller comparison. Section VIII concludes with directions and future work. The appendix IX collects detailed derivations, a KiCad drawing, gain sweep tables and the Algorithms used in this research.

II. Preliminaries / Background

This section establishes the theoretical foundation for the IBVS platform: the general IBVS framework, the feature extraction model and the interaction matrix mapping robot velocity to feature rates.

A. IBVS framework overview

Image-based visual servoing generates robot control commands directly from image-plane measurements, without explicit reconstruction of the 3-D target pose [1], [12]. The fundamental objective is to drive a vector of image features $\mathbf{s}(t) \in \mathbb{R}^k$ to a desired configuration $\mathbf{s}^* \in \mathbb{R}^k$ by commanding appropriate camera velocities. The feature error is defined as

$$\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^* \in \mathbb{R}^k, \tag{1}$$

The core kinematic relationship linking features rates to the camera spacial velocity $\xi_c = [\mathbf{v}_c^\top, \boldsymbol{\omega}_c^\top]^\top \in \mathbb{R}^6$ is

$$\dot{\mathbf{s}} = \mathbf{L}_s(\mathbf{s}, q)\, \xi_c, \tag{2}$$

[1] https://github.com/kkiewiet/Yolov5_ObjectDetection_redPH-and-SH-IBVS_controlLawComparisson_MecanumRover

where $\mathbf{L}_s \in \mathbb{R}^{k\times 6}$ is the interaction matrix (image Jacobian), dependent on the chosen features and the target configuration [1], [10]. Specifying the desired error dynamics $\dot{\mathbf{e}} = -\lambda \mathbf{e}$ with gain $\lambda > 0$, and substituting equation(1,2) yields

$$\xi_c = -\lambda \, \mathbf{L}_s^{+} \, \mathbf{e}, \tag{3}$$

where $\mathbf{L}_s^{+}$ denotes the Moore-Penrose pseudo-inverse. The specific form of $\mathbf{L}$ depends entirely on the choice of features $\mathbf{s}$, which is the subject of the following subsection.

B. Image Feature Extraction and the Pinhole Camera Model

The image formation process is described by the standard pinhole camera model [10], illustrated in Figure (1). Consider a target point with coordinates ${}^c\mathbf{p} = [{}^cX, {}^cY, {}^cZ]^\top$ expressed in the camera frame Σ_c, where the optical axis is aligned with the ${}^cZ$-axis and the image plane is located at focal distance f from the optical centre. Under the pinhole assumption, the perspective projection yields the image-plane coordinates

$$\mu \;=\; f_\mu \, \frac{{}^cx}{{}^cz}, \qquad \nu \;=\; f_\nu \, \frac{{}^cy}{{}^cz}, \tag{4}$$

where μ and ν are horizontal and vertical offsets from the image centre in pixels, and f_{px} is the focal length in pixels. For a camera with square pixels the horizontal and vertical focal lengths are approximately equal ($f_\mu \approx f_\nu$), allowing the use of a single calibrated focal parameter f_{px} throughout. The value of f_{px} is determined experimentally in Section V-C.

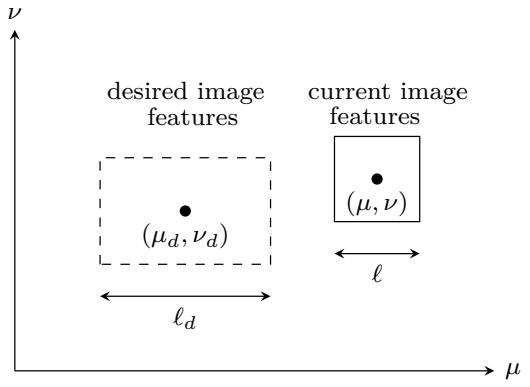

Fig. 1: Pinhole camera model showing the projection of a three-dimensional point $P({}^cX, {}^cY, {}^cZ)$ onto the image plane at $p_h = (\mu, \nu)$. The focal length is denoted by f, and the optical centre is indicated by o.

Equation (4) provides the two-dimensional image position (μ, ν) of the target, but depth along the optical axis ${}^cz$ is not directly observable from a single monocular view. A standard strategy is to exploit the known physical extent of the target [10]. If the target has a constant physical height h_{obj}, its apparent height in the image

$$\ell \;=\; \frac{f_\nu \, h_{\text{obj}}}{{}^cz}, \tag{5}$$

varies inversely with depth: ℓ grows as the camera approaches and shrinks as it recedes. Inverting this relationship yields the monocular depth estimate

$$\hat{Z} \;=\; \frac{f \, h_{\text{obj}}}{\ell}, \tag{6}$$

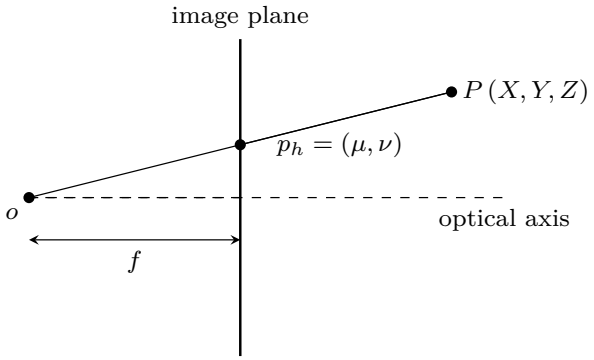

Fig. 2: Image plane representation of visual features. The dashed rectangle denotes the desired feature configuration (μ^*, ν^*, ℓ^*), while the solid rectangle represents the current feature state (μ, ν, ℓ). The apparent size ℓ serves as an estimate for camera-to-target distance, enabling monocular depth estimation through perspective scaling.

where f must be calibrated for the specific camera (Section V-C).[2] Together, (μ, ν, ℓ) characterise the complete apparent configuration as seen from the camera (Fig. 2). Figure (2) illustrates these three channels and the servoing task to drive the solid rectangle onto the dashed one.

C. General Planar Rover Simplification

For the platform considered here, a planar rover with a rigidly forward-facing camera, the vertical image coordinate ν is not independently controllable. Because the camera is mounted at fixed height and pitch, ν depends only on the target's height and the camera elevation, neither actuated by in-plane motion. Including ν would introduce a redundant constraint causing rank deficiency in the visual Jacobian. Therefore, a *reduced feature vector* $\mathbf{s} = [\mu, \ \ell]^\top$ is adopted, decoupling the servoing task into two independently regulable objectives: *horizontal alignment* ($\mu \to \mu^*$) and *distance regulation* ($\ell \to \ell^*$). With the feature vector established, we now derive the interaction matrix associated with this choice.

D. Interaction Matrix for Bounding-Box Features

In the general framework of Section II-A, the interaction matrix $\mathbf{L}_s \in \mathbb{R}^{k\times 6}$ relates k feature rates to the full camera spatial velocity $\xi_c \in \mathbb{R}^6$. The reduction to $\mathbf{s} = [\mu, \ell]^\top$ ($k = 2$) and to three actuated body DOF $\xi_b = [v_x, v_y, \omega_z]^\top$ collapses $\mathbf{L}_s$ into a 2×3 *visual interaction matrix* $\mathbf{J}_{\text{vis}}$ that maps robot body commands directly to feature rates. This composed form, adopted in [3], [5], [20] and used throughout the control laws of Section II-D, is obtained via the chain-rule decomposition

$$\dot{\mathbf{s}} \;=\; \underbrace{\mathbf{J}_f \, \mathbf{J}_c}_{=\mathbf{J}_{\text{vis}}} \, \xi_b, \tag{7}$$

[2]The apparent height of the bounding box (f_v) is chosen as the depth proxy rather than the width, because the physical height of the target remains approximately constant across viewing angles, whereas the apparent width varies with the object's orientation relative to the camera.

where $\mathbf{J}_f \in \mathbb{R}^{2\times 6}$ relates feature rates to the camera spatial velocity $\xi_c = [\mathbf{v}_c^\top, \omega_c^\top]^\top$, and $\mathbf{J}_c \in \mathbb{R}^{6\times 3}$ maps robot body commands to camera-frame velocities. The full algebraic derivation, based on [3], [20] with adaptations for the bounding-box height feature, is provided in Appendix IX.

Image Jacobian J_f: Differentiating the pinhole relations (4) and (5) with respect to time, for a static target under planar motion ($v_z = 0$, $\omega_x = \omega_y = 0$ in the robot frame):

$$\mathbf{J}_f(\mu,\ell,\hat{Z}) = \begin{bmatrix} -\dfrac{f_\mu}{\hat{Z}} & 0 & \dfrac{\mu}{\hat{Z}} & 0 & -\left(f_\mu + \dfrac{\mu^2}{f_\mu}\right) & 0 \\ 0 & 0 & \dfrac{\ell}{\hat{Z}} & 0 & -\dfrac{\ell\mu}{f_\mu} & 0 \end{bmatrix}. \tag{8}$$

Geometric coupling J_c: to evaluate (7), the camera spatial velocity ξ_c must be expressed in terms of the robot body twist ξ_b. This requires knowing the geometric relationship between the robot body frame Σ_r and the camera frame Σ_c. Figure (3) illustrates this frame assignment. Under planar motion, this rigid-body transformation gives the kinematic Jacobian:

$$\xi_c = \mathbf{J}_c\,\xi_b, \qquad \mathbf{J}_c = \begin{bmatrix} 0 & -1 & -d_c \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{6\times 3}, \tag{9}$$

encoding three geometric relationships between the rover body twist and the camera-frame velocity.

- $\dot{c}_x^c = -v_y - d_c\,\omega_z$: lateral camera motion arises from the rover's sideways velocity v_y, with an additional lever-arm contribution from the yaw rate ω_z acting through the camera offset d_c;
- $\dot{c}_z^c = v_x$: the camera depth rate equals the rover's forward velocity;
- $\omega_y^c = -\omega_z$: rover yaw maps directly to camera pan.

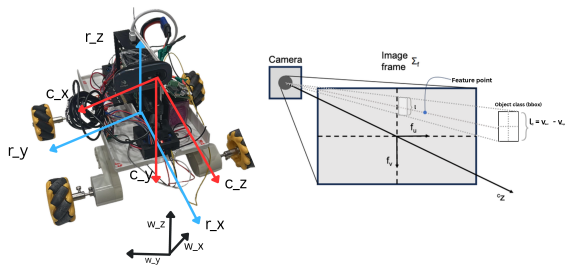

Fig. 3: Coordinate frame assignment for the mecanum rover platform, adapted from Ito et al. [3]. Three frames are shown: the world frame Σ_w (black), the robot body frame Σ_r (blue), and the camera frame Σ_c (red). The camera is rigidly mounted at a longitudinal offset d_c forward of the rover's centre of rotation and at height h_c. To conform with the standard optical convention (z-forward, x-right, y-down), the frame axes are mapped as follows: camera optical axis ${}^c z$ aligns with rover ${}^r x$, horizontal camera axis ${}^c x$ aligns with rover ${}^r y$, and vertical camera axis ${}^c y$ points along $-{}^r z$. Since the camera is fixed with no pitch or roll, only the rover's yaw angle ϕ couples the two frames, yielding the camera angular velocity ${}^w\omega_c = [0, 0, \dot{\phi}]^\top$.

Total visual Jacobian J_{vis}: Multiplying $\mathbf{J}_f$ (8) by $\mathbf{J}_c$ (9) yields the 2×3 visual interaction matrix that maps the robot body twist $\xi_b = [v_x, v_y, \omega_z]^\top$ directly to feature rates:

$$\mathbf{J}_{\text{vis}}(\mu,\ell,\hat{Z}) = \begin{bmatrix} \dfrac{\mu}{\hat{Z}} & \dfrac{f_\mu}{\hat{Z}} & f_\mu + \dfrac{\mu^2}{f_\mu} + \dfrac{f_\mu d_c}{\hat{Z}} \\ \dfrac{\ell}{\hat{Z}} & 0 & \dfrac{\ell\mu}{f_\mu} \end{bmatrix}, \tag{10}$$

where the depth $\hat{Z}$ is not measured directly but computed from the apparent height via (6).

III. Control Strategy

With the visual interaction matrix $\mathbf{J}_{\text{vis}}$ (10) established, the servoing task reduces to choosing a control law that maps the feature error $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ to a robot body twist ξ_b. Both controllers presented below operate on the same feature vector $\mathbf{s} = [\mu, \ell]^\top$ and the same Jacobian; they differ only in how they compute the commanded velocity.

A. Sugiura–Hashimoto PI-IBVS

The SH controller of Kota [5] and Hashimoto [12] refines the classical law (3) in two ways. First, the scalar gain λ is replaced by a positive-definite diagonal matrix $\kappa_P = \text{diag}(\kappa_{P,\mu}, \kappa_{P,\ell}) > 0$, assigning an independent convergence rate to each feature channel so that the lateral alignment (μ) and depth regulation (ℓ) dynamics can be tuned separately. Second, per-channel integral action is added to accumulate residual feature error over time, rejecting the steady-state offsets caused by motor dead zones, encoder quantisation, and modelling mismatch. Applying these modifications to the composed visual Jacobian $\mathbf{J}_{\text{vis}}$ derived in Section II-D yields the PI control law

$$\xi_b(t) = -\mathbf{J}_{\text{vis}}^{\dagger}(\mu,\ell)\left[\kappa_P\,\mathbf{e}(t) + \kappa_I \int_0^t \mathbf{e}(\tau)\,\mathrm{d}\tau\right], \tag{11}$$

where $\kappa_P = \text{diag}(\kappa_{P\mu}, \kappa_{P\nu}) > 0$, $\kappa_I = \text{diag}(\kappa_{I\mu}, \kappa_{I\ell}) \geq 0$, and $\mathbf{J}_{\text{vis}}^{\dagger}$ is the right Moore-Penrose pseudo-inverse $\mathbf{J}_{\text{vis}}^{\dagger} = \mathbf{J}_{\text{vis}}^\top \left(\mathbf{J}_{\text{vis}}\mathbf{J}_{\text{vis}}^\top\right)^{-1}$.

B. Port-Hamiltonian IBVS: Comparative Benchmark

The SH law (11) operates at the kinematic level: it prescribes velocity commands without accounting for inertia, dissipation, or dynamic coupling. The pH-IBVS controller of Muñoz-Arias et al. [20] addresses this by embedding the visual-servoing task within a port-Hamiltonian energy-based framework. The full derivation and stability proofs are given in [20]. This section summarises the structure and records the reduced form implemented on this platform.

The robot and visual-error dynamics are cast into the standard pH form

$$\dot{\mathbf{x}} = \left[\mathbf{J}(\mathbf{x}) - \mathbf{R}(\mathbf{x})\right]\frac{\partial H}{\partial \mathbf{x}} + \mathbf{G}(\mathbf{x})\,\mathbf{u}, \tag{12}$$

where $\mathbf{x}$ collects generalised momenta and visual error, $H(\mathbf{x})$ is the system Hamiltonian, $\mathbf{J}(\mathbf{x}) = -\mathbf{J}^\top$ encodes conservative interconnection (including gyroscopic and Coriolis

couplings), $\mathbf{R}(\mathbf{x}) \geq 0$ encodes natural dissipation, and $\mathbf{G}(\mathbf{x})$ is the input port matrix [20]. The control input $\mathbf{u}_{\text{pH}}$ is designed through Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC) [17]. The closed-loop system possesses the desired Hamiltonian

$$H_d \;=\; \tfrac{1}{2}\,\mathbf{e}^{\top}\kappa_P\,\mathbf{e} \;+\; \tfrac{1}{2}\,\mathbf{z}^{\top}\kappa_I^{-1}\,\mathbf{z} \;\geq\; 0, \tag{13}$$

where $\mathbf{z}$ is the leaky integral state. By construction, H_d serves as a Lyapunov function: Theorem 1 of [20] guarantees that $\dot{H}_d \leq 0$ along closed-loop trajectories, ensuring asymptotic convergence to $\mathbf{s}^*$. In principle, logging H_d in real time provides a scalar certificate of controller performance; this capability is not exploited in the present experiments but is available in the open-source implementation for future use.

C. Reduced-pH

The general law of [20] is derived for a full six-DOF mechanical pH system. Three simplifications reduce it to the form implemented on this platform:

a) Flat floor ($V \equiv 0,\ \Gamma \approx 0$): Motion is confined to the horizontal plane, so the gravitational potential energy vanishes and gyroscopic forces are negligible. The interconnection matrix $\mathbf{J}(\mathbf{x})$ consequently carries no terrain-coupling terms.

b) Velocity-level operation ($\mathbf{M} = \mathbf{I}$): The complete pH law requires the inertia matrix $\mathbf{M}(\mathbf{q})$, the Coriolis matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, and their estimates in real-time. At the velocity level with constant unit inertia, these terms vanish, reducing the implemented control law to

$$\xi_b(t) \;=\; -\mathbf{J}_{\text{vis}}^{\dagger}\Big[\kappa_P\,\mathbf{e}(t) + \kappa_I\,\mathbf{z}(t)\Big] \;-\; \mathbf{J}_{\text{vis}}^{\dagger}\,\kappa_d\,\mathbf{J}_{\text{vis}}\,\xi_b(t), \tag{14}$$

where $\kappa_d = \text{diag}(\kappa_{d\mu}, \kappa_{d\ell}) \geq 0$ is the damping injection gain and the leaky integral state evolves as $\dot{\mathbf{z}} = \mathbf{e} - \lambda_{\text{leak}}\,\mathbf{z}$.

c) Reduced feature set: The vertical image feature ν evolves passively on a flat floor with a fixed camera; it is not included in $\mathbf{e}$ and enters the interaction matrix only as a parameter, as established in Section II-C.

When $\kappa_d = \mathbf{0}$ and $\lambda_{\text{leak}} = 0$, equation (14) algebraically to equation (11), confirming that the pH controller generalizes the kinematic baseline.

D. Energy-Based Comparison Methodology

A principled comparison of the two control laws requires a metric that captures both task performance and control effort. Following Ma et al. [21], who compare a port-Hamiltonian force-impedance controller against a classical Euler-Lagrange counterpart on the PERA manipulator, we adopt the cumulative L_2 actuation cost

$$\mathcal{E} \;=\; \int_0^{T_c} \|\xi_b(t)\|^2\,\mathrm{d}t, \tag{15}$$

evaluated up to the convergence time T_c.

In the velocity-level setting where $\mathbf{M} = \mathbf{I}$, the squared norm $\|\xi_b\|^2$ is proportional to commanded kinetic energy. The integral in (15) therefore penalises both the magnitude and the duration of actuation, rewarding controllers that drive the features to their setpoints with minimal cumulative effort. This property makes $\mathcal{E}$ particularly suitable for comparing controllers that achieve similar convergence times but differ in transient behaviour.

Because both controllers share identical inner-loop dynamics and hardware, the relative ranking of $\mathcal{E}$ constitutes a valid comparison metric. We note, however, that $\|\xi_b\|^2$ quantifies commanded actuation intensity rather than true electrical energy consumption; this would require integration of motor currents, which lies outside the scope of this study.

IV. Software System Architecture

The software implements a hierarchical control architecture separating high-level perception from low-level actuation (Fig. 4), with Algorithms (1–4) in Appendix IX. The three cascaded layers are described below.

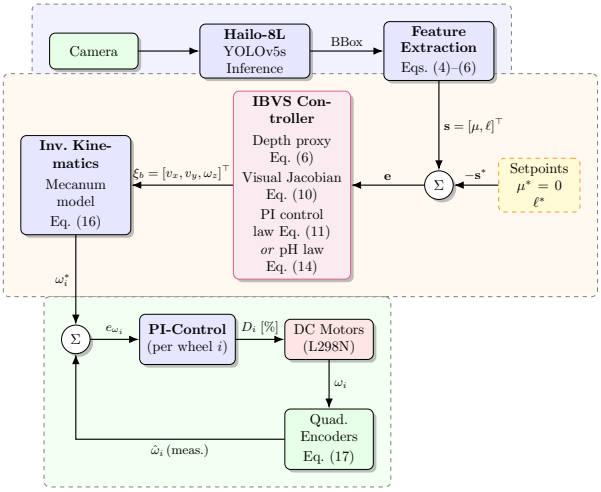

Fig. 4: Hierarchical system architecture. (1) Vision Pipeline (30 Hz) extracts features $\mathbf{s}$; (2) IBVS Controller computes body twist $\mathbf{v}_c$; (3) Inner-Loop (50 Hz) regulates wheel speeds $\boldsymbol{\omega}$.

A. Vision pipeline

The vision pipeline runs at 30 Hz using a Logitech C270 webcam. Frames are preprocessed and fed to the Hailo-8L accelerator running YOLOv5 inference on a COCO-trained model supporting multiple object classes. Detections are filtered by class ("bottle") and confidence threshold (> 0.4), with hysteresis-based filtering to reject transient noise.

The visual features μ, ℓ, and depth estimate $\hat{Z}$ are extracted from the detected bounding box as described in Algorithm 1. The horizontal feature μ and vertical feature ℓ are computed from the box center coordinates, while depth $\hat{Z}$ is estimated from the apparent height.

B. Control laws

The control layer computes the feature error $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$, evaluates J_{vis} at the current feature state and depth estimate, and applies either the SH law (11) or the reduced pH law (14) to generate the body twist ξ_b

C. Inner wheel loop

The inner wheel loop operates asynchronously at 50Hz, responsible for translating the commanded body twist ξ_b into individual wheel velocity setpoints ω^* via the inverse kinematics model (Eq. 16). A proportional-integral (PI) controller regulates each wheel's speed using quadrature encoder feedback, adjusting the PWM duty cycle D_i.

1) Mecanum Rover Kinematics: Following Lynch and Park [22], the mapping from $\xi_b = [v_x, v_y, \omega_z]^\top$ to wheel angular velocities $\omega = [\omega_1, \omega_2, \omega_3, \omega_4]^\top$ is

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(L_x + L_y) \\ 1 & 1 & (L_x + L_y) \\ 1 & 1 & -(L_x + L_y) \\ 1 & -1 & (L_x + L_y) \end{bmatrix} \xi_b, \tag{16}$$

where r is the wheel radius, L_x and L_y are the half-length and half-width of the wheelbase, and wheel numbering follows the convention 1=FL, 2=BL, 3=FR, 4=BR (Figure 5).

2) Wheel Velocity Feedback: The angular velocity of each wheel is estimated from the quadrature encoder output. At each control cycle (period $\Delta t = 0.02\,\text{s}$), the inner loop reads the cumulative encoder count pos_i and computes the count change Δpos_i since the previous cycle. The wheel angular velocity follows as

$$\hat{\omega}_i = \frac{\Delta \text{pos}_i}{\Delta t} \cdot \frac{60}{\text{CPR}_{\text{wheel}}}, \tag{17}$$

where $\text{CPR}_{\text{wheel}}$ is the number of encoder counts produced by one full revolution of the wheel shaft. Here, $\text{CPR}_{\text{wheel}}$ is a calibrated constant determined in Section V-B.

V. Experimental Setup

A. Hardware architecture

The experimental platform is based on the Rover-5 tracked chassis [23], modified with mecanum wheels to enable omnidirectional holonomic motion. Key specifications are summarised in Table I; the electrical architecture is illustrated in Figure 6 and the KiCad schematics are also given in Appendix Figure (19).

TABLE I: Experimental Platform Specifications

Component	Specification
Chassis	Rover-5 (modified)
Mass	3.0 kg (total)
Gear ratio	86.8:1
Encoder	$\approx$ 333 CPR (quadrature)
DC Motor	11 V, 2.5 A stall
Camera	Logitech C270 (720p, 55° FoV)
Computer	Raspberry Pi 5 (8 GB) + Hailo-8L
Battery	4S LiPo (11.2 V nominal)
Drivers	2× L298N Dual H-Bridge

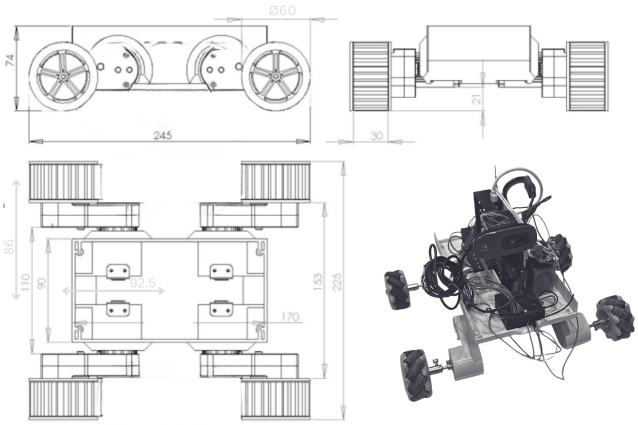

Fig. 5: Rover-5 chassis modified with mecanum wheels ($r = 0.06$ m). Kinematic parameters (Eq. (16)): $\ell_x = 0.093$ m (half-length), $\ell_y = 0.087$ m (half-width). Camera offset: $d_c = 0.09$ m forward, $h_c = 0.08$ m height from robot center, optical axis (${}^c z$ aligned with ${}^r x$-axis.

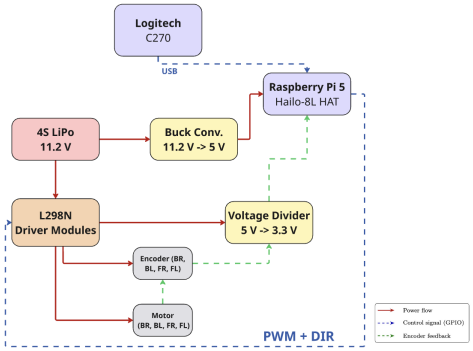

Fig. 6: System integration schematic illustrating the electrical architecture and signal flow between major components. Power distribution (red): The 4S LiPo battery (11.2 V) supplies the L298N motor drivers directly and feeds the Raspberry Pi 5 via a Pololu buck converter (11.2 V $\rightarrow$ 5 V). Control signals (blue): PWM and direction commands are sent from Raspberry Pi GPIO to the motor drivers. Encoder feedback (green): Quadrature encoder signals from each wheel are level-shifted from 5 V (L298N VCC) to 3.3 V via voltage dividers for GPIO compatibility. The Logitech C270 webcam interfaces via USB, while the Hailo-8L accelerator is mounted as a HAT module. 4 DC motors included, see Table I for specifications.

B. Encoder Characterization

Accurate velocity estimation via (17) requires a reliable value of $\text{CPR}_{\text{wheel}}$. The Rover-5 datasheet specifies $\text{CPR}_{\text{wheel}} \approx 333$ counts/rev [23], but the exact value can deviate. Therefore, an independent characterization is performed to verify this specification on each wheel.

The encoder resolution was measured by counting the total number of encoder ticks accumulated over a known number of complete wheel revolutions:

$$\text{CPR}_{\text{wheel}} = \frac{\Delta C}{N_{\text{rev}}}, \tag{18}$$

where ΔC is the total encoder count change and N_{rev} is the number of full revolutions. A visible mark on the wheel rim was aligned with a fixed reference on the chassis and the encoder count was reset to zero. Then, the wheel was rotated at low speed, and the cumulative count was recorded each time the mark returned to the reference. Multiple revolutions were

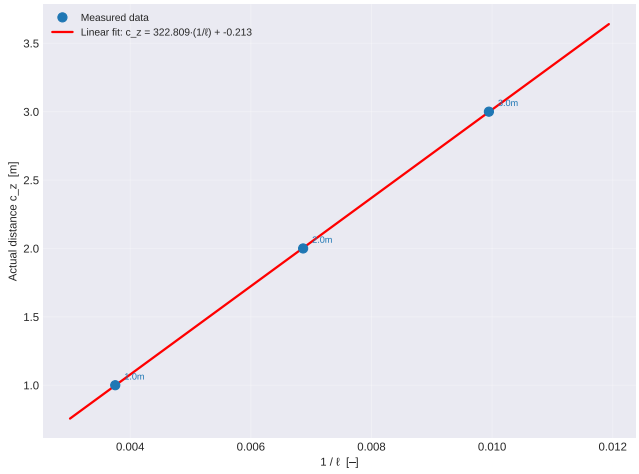

Fig. 7: Calibration of the vertical focal length f_y. Linear regression ($R^2 = 1.0$) of actual distance versus $1/\ell$ yields a slope of 322.81 m px (solid line), corresponding to $f_y = 1403.5$ px. The intercept of -0.213 m reflects a systematic offset in the distance reference.

collected per wheel to obtain a per-revolution breakdown and verify consistency.

Table II reports the cumulative encoder counts after each revolution for all four wheels. The per-revolution increments $\delta C_k = C_k - C_{k-1}$ (with $C_0 = 0$) deviate by a few counts around the mean, which is the lack of visual alignment.

TABLE II: Cumulative encoder counts C_k after each complete wheel revolution, and the resulting mean $\mathrm{CPR}_{\mathrm{wheel}}$. Per-revolution increments $\delta C_k = C_k - C_{k-1}$ are given in parentheses.

	FL (wheel 1)	BL (wheel 2)	FR (wheel 3)	BR (wheel 4)
C_1 (δC_1)	331 (331)	330 (330)	340 (340)	341 (341)
C_2 (δC_2)	664 (333)	670 (340)	677 (337)	670 (329)
C_3 (δC_3)	1001 (337)	1000 (330)	1001 (324)	1000 (330)
N_{rev}	3	3	3	3
Total ΔC	1001	1000	1001	1000
CPR (mean)	**333.7**	**333.3**	**333.7**	**333.3**

The measured resolution is 333.3–333.7 counts/rev across all four wheels, in close agreement with the datasheet value.

C. Focal Length Calibration

The depth model (6) requires the vertical focal length f_y of the camera, expressed in pixels. Rearranging the pinhole relation into the linear form $Z = (f_y h_{\mathrm{obj}}) \cdot (1/\ell)$ shows that f_y can be recovered from the slope of a least-squares fit of measured distance Z versus inverse apparent height $1/\ell$.

An iterative calibration procedure was carried out to determine f_y reliably. In each iteration, the target bottle ($h_{\mathrm{obj}} = 0.23$ m) was placed at three known distances ($Z = 1.0, 2.0, 3.0$ m) and the mean bounding-box height $\bar{\ell}$ was recorded in pixels at each position. Successive iterations refined the camera placement and detection filtering until the regression residuals stabilised.

The final calibration result is shown in Figure 7. A linear least-squares fit of the measured distance Z against the inverse apparent height $1/\ell$ yields a slope of 322.81 m px and an intercept of -0.213 m, with a goodness of fit $R^2 = 1.0$. The non-zero intercept indicates a systematic offset between the reference plane used for distance measurement and the optical centre of the camera. The vertical focal length follows as $f_y = \mathrm{slope}/h_{\mathrm{obj}} = 322.81/0.23 = 1403.5$ px.

VI. Experimental Protocol

All experiments are conducted in a controlled indoor environment on a 2×1.2 m testing area (Figure 8). A target bottle ($h_{\mathrm{obj}} = 0.23$ m) is placed at prescribed lateral offsets from the rover centreline at an initial distance of approximately 2.06 m. Each configuration is repeated at least three times to verify repeatability (except for the gain sweeps).

A. Performance Metrics

Closed-loop performance is evaluated per feature channel $i \in \{\mu, \ell\}$ using four indicators:

- **Rise time** T_r: the earliest time at which $|e_i|$ drops below 10% of its initial peak. The reference peak is taken as the maximum of $|e_i|$ over the first quarter of the detection window, to avoid inflation by later transients.
- **Settling time** T_s: the earliest time after which $|e_i|$ remains within a convergence band for at least five consecutive frames. The bands are ± 0.01 for both μ and ℓ.
- **Mean absolute error** (MAE): the time-averaged $|e_i|$ over the entire detection window, including the transient.
- **Steady-state error** (SSE): identical to MAE but restricted to samples at $t \geq T_s$, capturing only the converged portion. Returns (—) if the controller does not settle.

B. Experimental Campaign

The campaign proceeds in five stages, each building on the previous. Table III provides a compact overview of the stages, the corresponding result figures in Section VII, and the appendix tables containing the full numerical data.

TABLE III: Experimental campaign overview.

Stage	Description	Figures	Tables
1	Baseline repeatability	9–10	IV
2	Gain tuning (P, then PI)	11–12	V–VI
3	Reduced pH equivalence	13	VII
4	Reduced pH Damping sweep	14	VIII
5	Comprehensive comparison	15–16	IX

1) **Baseline repeatability** The SH PI-IBVS law (11) is run from symmetric ± 0.5 m lateral offsets in holonomic mode (v_y enabled). Three trials per side establish the within-side standard deviation of all four metrics, confirming that the platform produces repeatable closed-loop trajectories suitable for comparative evaluation.
2) **Gain tuning** Proportional gains are swept first (with $\mathbf{K}_I = \mathbf{0}$). Following Lynch and Park [22], the proportional gains are selected to achieve the fastest settling without overshoot, corresponding to a near-critically-damped first-order error response. Integral gains are then introduced incrementally using the standard heuristic $K_I \approx 0.1\,K_P$ [22], and adjusted to minimise the steady-state mean (SSM) without inducing integral

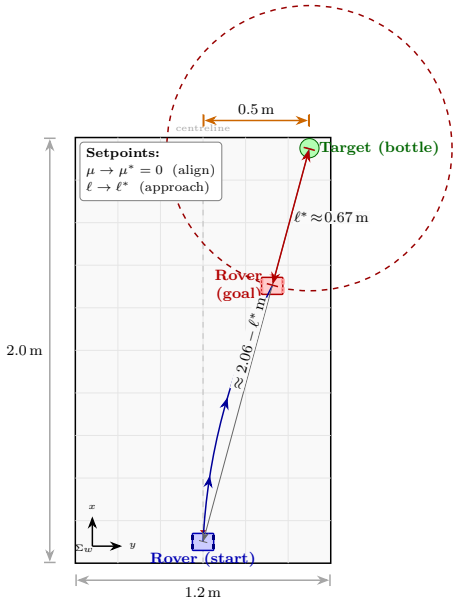

Fig. 8: Experimental test area (2×1.2 m) with rover at centre start position and target bottle ($h_{\text{obj}} = 0.23$ m) at ≈ 2.06 m range. $\ell^* = 460$ Blue curve: representative trajectory from lateral offset to centred alignment.

windup. The combination that best balances T_r, T_s, and SSE is selected for all subsequent stages.

3) **Reduced pH equivalence** The pH law (14) is evaluated with $\mathbf{K}_d = \mathbf{0}$ and $\lambda_{\text{leak}} = 0$, which algebraically recovers the SH law. Three runs per controller verify that both implementations produce statistically indistinguishable trajectories, confirming code-level equivalence before activating the pH-specific terms.
4) **Damping Sweep** The pH law (14) is evaluated with sweeping $\mathbf{K}_d \neq \mathbf{0}$ and fixed value $\lambda_{\text{leak}} = 0.03$. Performance is assessed on T_s, SSE, and the cumulative L_2 actuation cost $\mathcal{E}$ (15).
5) **Comprehensive comparison** The full pH controller, with damping injection and leaky integration active, is compared against the SH PI-IBVS baseline. The pH-specific gains ($\mathbf{K}_d$, λ_{leak}) were selected from preliminary sweeps. Following Muñoz-Arias et al. [20] and Ma et al. [21], both controllers are matched for convergence performance and compared on five metrics: feature convergence, feature-space trajectory, commanded body velocities and cumulative L_2 actuation cost $\mathcal{E}$ (15) in the (μ, ℓ) plane.

VII. Experimental Results

This section reports the experimental findings obtained following the five-stage protocol defined in Section VI. Each subsection corresponds to one campaign stage: baseline repeatability, gain tuning, reduced pH equivalence verification, and the comprehensive controller comparison.

A. Baseline Repeatability

Three trials per side were conducted from ± 0.5 m lateral offsets under identical SH gains with holonomic motion (v_y enabled). Full numerical results are collected in Appendix Table IV.

1) Holonomic Mode Repeatability: The holonomic kinematics permit simultaneous μ correction through yaw and lateral translation, exploiting the structural decoupling established in Section II-D. As shown in Figure (9) and Table IV, left-side trials yield $T_r(\mu) = 2.564 \pm 0.032$ s and $T_s(\mu) = 7.736 \pm 1.599$ s, while right-side trials yield $T_r(\mu) = 3.897 \pm 0.113$ s and $T_s(\mu) = 4.291 \pm 0.266$ s. The depth channel converges with $T_s(\ell) = 18.414 \pm 2.143$ s (left) and $T_s(\ell) = 12.596 \pm 1.239$ s (right).

Within each side, per-trial standard deviations remain below 2 s across all metrics, confirming acceptable run-to-run repeatability. The approximately 6 s left–right asymmetry in the ℓ settling time originates from two sources: initial manual placement tolerances and an asymmetric battery mass distribution that biases the rover's centre of gravity toward the left. During left-offset trials, the rover advances nearly straight toward the target, prioritizing ℓ due to its high error and keeping μ small throughout the transient. During right-offset trials, the rover initially commands a large forward velocity v_x driven by the depth-channel error. This aggressive approach is a consequence of the $\ell^2/(f_v h_{\text{obj}})$ nonlinearity in the visual Jacobian, which amplifies v_x when the rover is far from the target and ℓ is small (see Section VIII for a detailed analysis). The resulting forward surge temporarily increases μ through the yaw coupling in column 3 of $\mathbf{J}_{\text{vis}}$ (10), pushing the bounding box toward the image boundary before a sharp compensating yaw re-centres the target. Combined with the platform's mechanical left–right asymmetry (off-centre battery mass, imperfect wheel alignment), this coupling between the ℓ- and μ-transients produces the curved trajectories visible in Figure (10). Despite this systematic offset, within-side consistency across three runs validates the platform as a comparative controller testbed.

B. Gain Tuning

To avoid "favouring" the left-side asymmetry, all subsequent experiments place the target on the rover's right. With integral action disabled ($\mathbf{K}_I = \mathbf{0}$), proportional gains were varied across $K_{P,\mu} \in [1.5, 30.0]$ and $K_{P,\ell} \in [0.2, 0.6]$. The full parameter sets and metrics are listed in Appendix Tables V–VI and illustrated in Figures 11–12.

1) Proportional Gains: Low proportional gains ($K_{P,\mu} < 1.5$) produced sluggish yaw responses with settling times exceeding 25 s, resulting in impractical configurations. Progressively increasing the gains to $K_{P,\mu} > 12$, $K_{P,\ell} > 0.4$ (gain4+) reduced $T_s(\mu)$ from > 20 s to ≈ 4.4 s and $T_s(\ell)$ to ≈ 12 s. However, raising $K_{P,\mu}$ beyond this point introduced overshoot and oscillatory behaviour, while setting $K_{P,\ell} \geq 0.7$ over-prioritised depth regulation, causing the lateral error to grow until the target exited the camera's field of view. Conversely, under-prioritising ℓ (gain6, $K_{P,\ell} = 0.2$) achieved faster μ centering ($T_s(\mu) = 3.925$ s) at the cost of prohibitively slow depth convergence ($T_s(\ell)$ did not settle within the 30 s window).

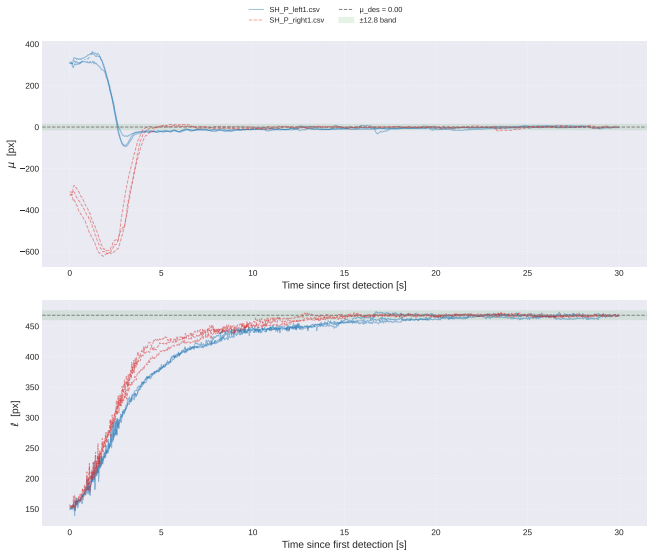

Fig. 9: Baseline repeatability (SH P-IBVS, holonomic mode). Three trials from $+0.5\,\mathrm{m}$ offset (solid) and three from $-0.5\,\mathrm{m}$ offset (dashed). (a) μ convergence. (b) ℓ convergence.

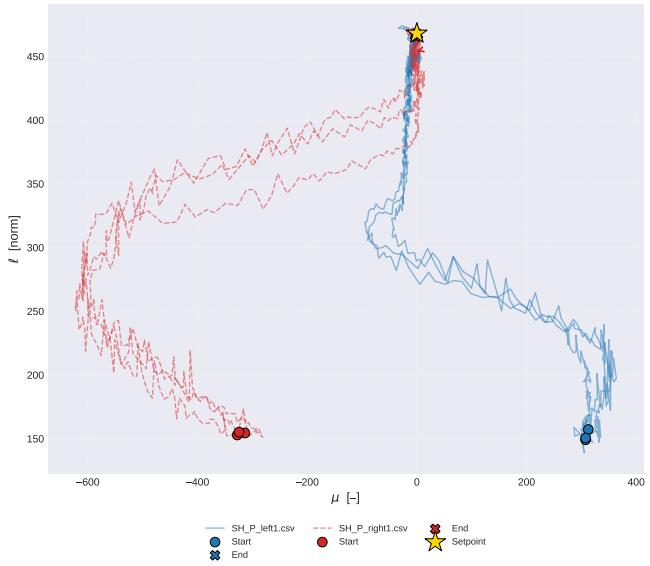

Fig. 10: Overhead view of the holonomic SH P-IBVS trajectories, complementing Figure 9. The near-identical starting positions and the paths converging to the setpoint (star) are shown for both offset sides.

During aggressive forward motion, ℓ showed transient spikes at relatively high gains, which is attributed to partial bounding-box occlusion.

Based on these sweeps, the proportional gains that achieve the fastest settling without overshoot were identified (gain3_3), yielding the working-point $\kappa_{P,\mu} = 17.0$, $\kappa_{P,\ell} = 0.5$.

2) Integral Gains: Integral action was introduced incrementally using the standard heuristic $K_I \approx 0.1\,K_P$ [22] as a starting point. Table VI and Figure (12) present the PI tuning results.

Adding $K_{I,\mu} = 0.003$ (gain3_1) reduced the steady-state error from $\mathrm{SSE}(\mu) = 3.317$ *px* to 2.664 *px*. Increasing to $K_{I,\mu} = 0.005$ (gain3_2), however, elevated $\mathrm{SSE}(\mu)$ back to

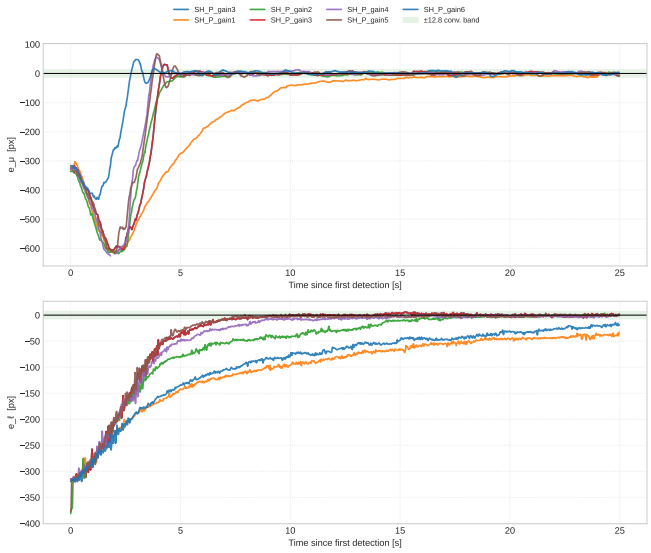

Fig. 11: Proportional gain sweep: μ and ℓ error trajectories under varying gain combinations (Table V).

3.428 *px* without a corresponding improvement in settling time, indicating already integral overshoot.

In the depth channel, raising $K_{I,\ell}$ to 0.005 (gain3_4), increased in $\mathrm{SSE}(\mu)$ from 1.914 px to 6.900 px, indicating no integral term is needed for ℓ, but surprisingly lower $T_s(\mu,\ell) = 3.925$ s. Activating integral action in both channels simultaneously with $K_{I,\mu} = 0.002$, $K_{I,\ell} = 0.001$ (gain3_5) achieved a steady-state error comparable to the proportional-only baseline while slightly reducing $T_s(\mu)$ at the cost of a 2 s increase in $T_s(\ell)$.

The PI configuration, $K_{P,\mu} = 17.0$, $K_{P,\ell} = 0.5$, $K_{I,\mu} = 0.002$, $K_{I,\ell} = 0.001$ is selected, which represents a baseline for subsequent pH controller experiments. While pure proportional control achieved better performance, integral action is necessary for the port-Hamiltonian formulation to enable the leaky integrator and integral action. The chosen gains $K_{I,\mu}, K_{I,\ell}$ are intentionally small to balance integral action with transient performance and stability margins.

C. Reduced pH Equivalence Verification

With $\mathbf{K}_D = \mathbf{0}$ and $\lambda_{\text{leak}} = 0$, the pH control law (14) algebraically reduces to the SH law (11). To verify code-level equivalence, three runs were conducted per controller under matched proportional gains ($K_{P,\mu} = 17.0$, $K_{P,\ell} = 0.5$) with all integral, damping, and leaky terms disabled.

As shown in Figure 13 and Table VII, the feature-space trajectories of both implementations overlap closely. The mean rise and settling times differ by less than 0.4 s across all metrics, and the standard deviations remain below ≈ 1 s in both channels. The steady-state errors ($\mathrm{SSE}(\mu) \approx 3.65$ px, $\mathrm{SSE}(\ell) \approx 2.5$ px) and the cumulative actuation cost ($E \approx 6$ (m/s)^2·s) are statistically indistinguishable between the two implementations. This equivalence confirms that the pH code path introduces no unintended numerical deviations and that any performance differences observed after activating the

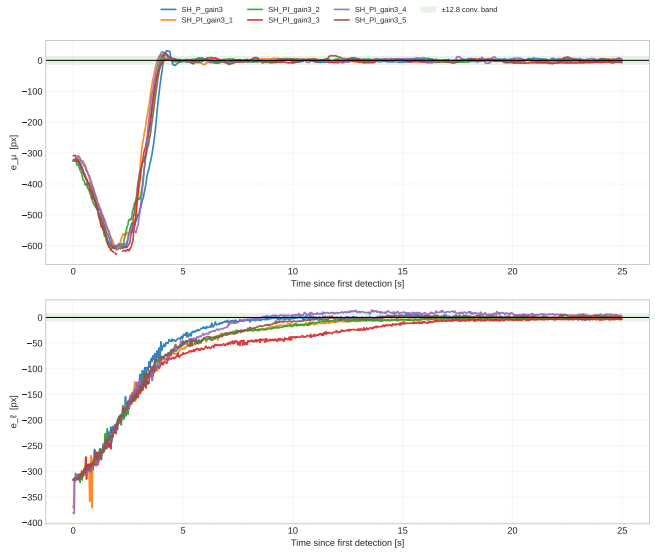

Fig. 12: Integral gain sweep: μ and ℓ error trajectories under varying integral gain combinations (Table VI).

pH-specific terms can be attributed to damping injection, leaky integration or small offset changes.

Fig. 13: Feature-space trajectories of the SH PI-IBVS (blue) and the reduced pH-IBVS (red) under matched gains. The target is placed at +0.5 m lateral offset, corresponding to $\mu \approx -300$ px; the desired $\ell^* = 471$ px is indicated by the star.

D. Damping Injection and Leaky Integral Activation

Building on the verified reduced pH controller, the two mechanisms absent are the damping injection ($\mathbf{K}_D$) and leaky integration (λ_{leak}) that are activated now. The leaky coefficient was fixed at $\lambda_{\text{leak}} = 0.03$ throughout, while damping gains were swept across $K_{D,\mu} \in [0, 0.5]$ and $K_{D,\ell} \in [0, 0.3]$. The results are summarised in Table VIII and Figure 14.

Introducing damping in the μ channel alone ($K_{D,\mu} = 0.3$, gain1) slightly increased the settling time of both channels compared to the undamped reference (redPH), but produced a smoother yaw trajectory, reducing the peak angular velocity from 2.877 rad/s to 2.337 rad/s and the cumulative actuation cost from $E = 6.551$ to 4.310 (m/s)^2·s. Increasing $K_{D,\mu}$ further to 0.5 (gain3) lowered E to 3.382 (m/s)^2·s, respectively, each time reducing the maximum yaw rate while accepting a moderate increase in $T_s(\ell)$.

Damping the ℓ channel was investigated more conservatively, since the depth response was already converging smoothly in the undamped case. However, when $K_{D,\mu}$ is large, the μ-channel damping can delay the initial centering phase and, through the coupling in J_{vis}, induce transient overshoot in μ as the rover approaches. Adding a small $K_{D,\ell} = 0.1$ alongside $K_{D,\mu} = 0.5$ (gain3) mitigated this coupling, yielding the lowest SS-MAE(μ) = 2.586 px and the lowest overall actuation cost $E = 3.382$ (m/s)^2·s. Increasing $K_{D,\ell}$ to 0.3 (gain4) raised E again to 4.696 (m/s)^2·s due to excessive suppression of the forward velocity transient.

The trajectory plots in Figure (14) illustrate this trade-off: higher damping steers the feature-space path relatively closer to a curved smooth line from the initial condition to the setpoint, representing the minimum-energy trajectory, but at the cost of a longer convergence time.

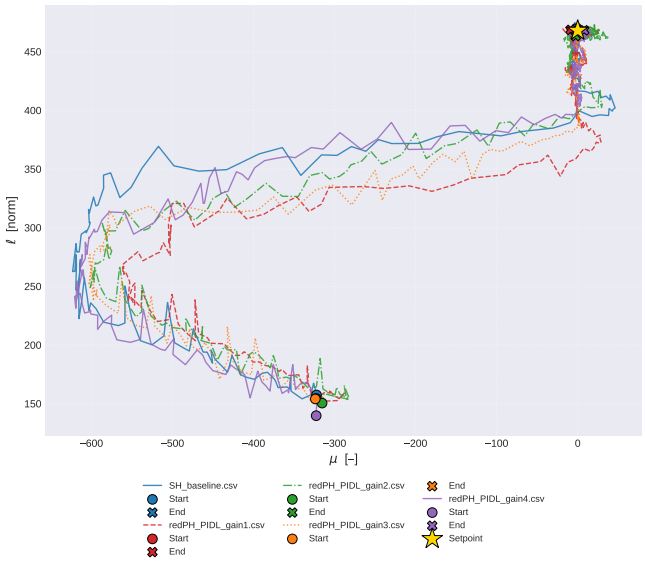

Fig. 14: Feature-space trajectories under varying damping configurations. Baseline in yellow. Increasing K_D steers the trajectory toward the straight-line minimum-energy path at the expense of longer convergence.

E. Comprehensive Controller Comparison

Two controller configurations were selected for the final comparison: the SH P-IBVS baseline (*SH_P_gain3_2*) and the full pH controller with combined damping and leaky integrator (*redPH_PIDL_gain3*). Both controllers share the proportional and integral gains $K_{P,\mu} = 17.0$, $K_{P,\ell} = 0.5$, $K_{I,\mu} = 0.002$, $K_{I,\ell} = 0.001$; the pH variant additionally uses $K_{D,\mu} = 0.5$, $K_{D,\ell} = 0.1$, and $\lambda_{\text{leak}} = 0.03$. Each controller was run three times, and the comparison metrics are reported in Table IX and Figures 15–16.

1) Feature Convergence and Trajectory (Figs. 15–16): The feature-space trajectories and error time histories of both controllers are qualitatively similar, as expected from

the shared proportional and integral structure. Both follow a curved path that initially over-prioritises the ℓ direction, driven by the large forward velocity commanded by the depth error, before correcting μ through an aggressive yaw that overshoots the $\mu = 0$ line. The pH controller, however, traces a slightly smoother path toward the setpoint, closer to the straight-line minimum-energy trajectory connecting the initial condition to (μ^*, ℓ^*), as the damping term moderates the overshoot.

This difference is visible in the error time histories (Fig. 16). In the μ channel, both controllers achieve comparable settling times ($T_s(\mu) = 4.006 \pm 0.313$ s for pH versus 4.421 ± 0.110 s for SH) and steady-state errors (SS-MAE$(\mu) = 3.256 \pm 0.669$ px versus 3.588 ± 0.376 px). In the ℓ channel, the pH controller converges more slowly ($T_s(\ell) = 11.931 \pm 2.185$ s versus 7.149 ± 0.457 s) but reaches a lower steady-state error (SS-MAE$(\ell) = 1.713 \pm 0.118$ mm versus 2.722 ± 0.677 mm), as the damping suppresses the initial forward velocity peak and the overshoot visible in the SH response. The per-controller standard deviations across three runs remain small for both configurations (Table IX), confirming that the observed differences are systematic rather than run-dependent.

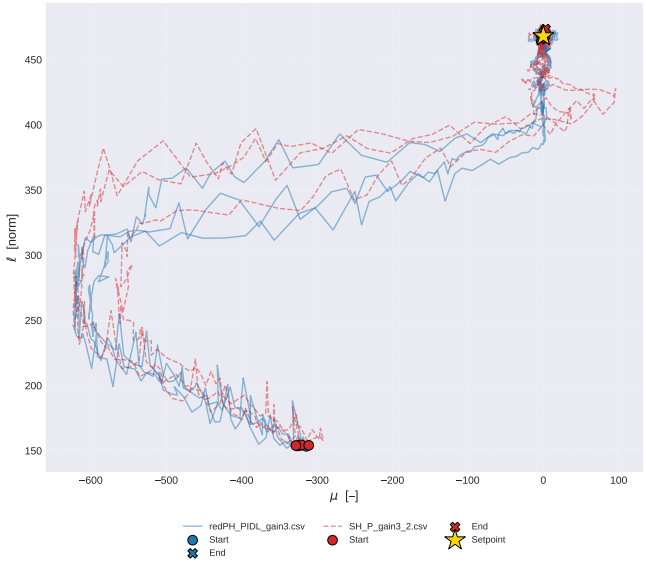

Fig. 15: Feature-space trajectories of the SH PI-IBVS (blue) and full pH-IBVS (red) controllers over three runs each. The star marks the desired setpoint (μ^*, ℓ^*).

2) Commanded Body Velocities (Fig. 17): The commanded velocity profiles of both controllers are qualitatively similar, as expected from the shared proportional and integral structure. The key difference lies in transient smoothness: the pH controller produces fewer oscillations in both ω_z and v_x during the initial centering phase. The peak yaw rate is reduced from $|\omega_z|_{\max} = 2.622 \pm 0.132$ rad/s (SH) to 1.949 ± 0.109 rad/s (pH), a 25.7% reduction that is directly attributable to the damping injection term $-\mathbf{J}_{\text{vis}}^{\dagger}\mathbf{K}_D\mathbf{J}_{\text{vis}}\boldsymbol{\xi}_b$. The maximum forward velocity remains essentially unchanged ($|v_x|_{\max} \approx 0.765$ m/s for both).

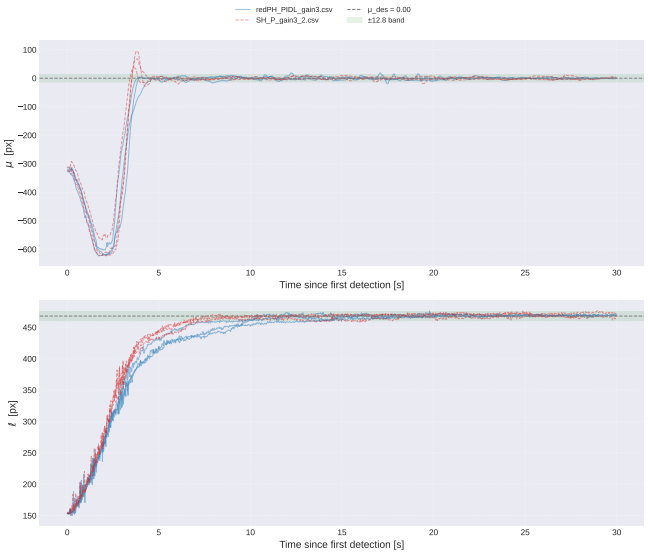

Fig. 16: Feature error convergence for both controllers: (a) μ channel, (b) ℓ channel. The pH controller exhibits a smoother transient with reduced overshoot.

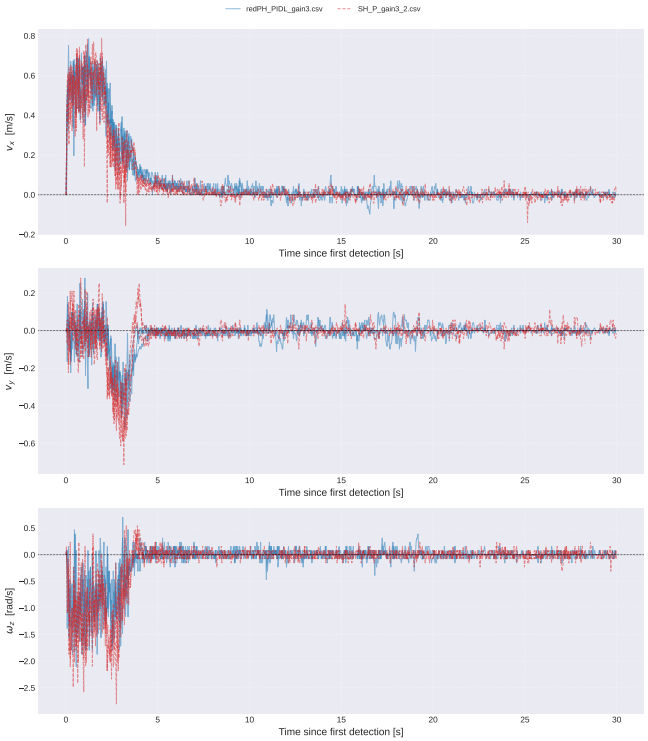

Fig. 17: Commanded body velocities v_x, v_y and ω_z for both controllers. The pH controller exhibits reduced oscillatory behaviour.

3) Cumulative L_2 Actuation Cost (Fig. 18): The cumulative L_2 actuation cost, defined in (15), provides a scalar summary of control effort. The pH controller achieves $E = 4.117 \pm 0.512$ (m/s)^2·s, representing a 34.4% reduction compared to the SH baseline at $E = 6.271 \pm 0.581$ (m/s)^2·s. This reduction is consistent with the lower peak yaw rate and the damped velocity profile mentioned above. The standard deviation between runs is small for both controllers, indicating that the efficiency gain is repeatable.

Figure 18 shows both the instantaneous sum of commanded body velocity magnitudes (Sum of Figure17) and the cumulative integral. The pH curve rises more slowly throughout the transient, confirming that the controller distributes its actuation energy more evenly rather than concentrating it in a brief, aggressive correction phase.

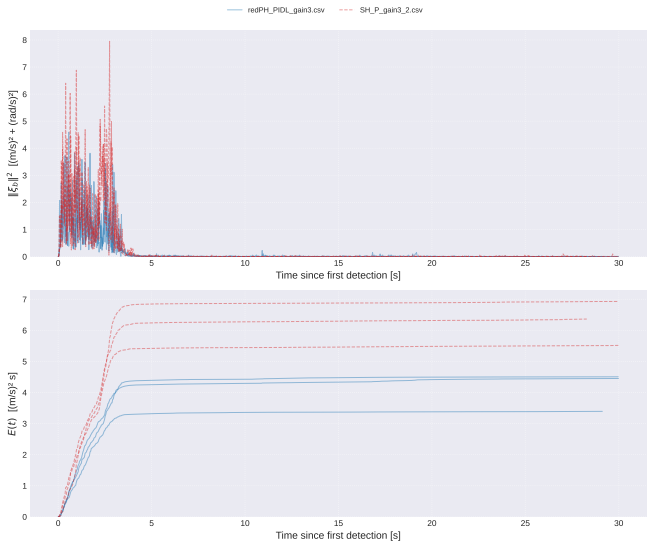

Fig. 18: (a) Instantaneous sum of commanded body velocity magnitudes ($|v_x| + |v_y| + |\omega_z|$). (b) Cumulative L_2 actuation cost $E(t) = \int_0^t \|\xi_b\|^2 \, d\tau$ for both controllers.

VIII. Concluding Remarks & Future Work

The experimental results confirm that even a velocity-level reduction of the pH-IBVS framework provides measurable benefits over the established Sugiura-Hashimoto PI-IBVS baseline: a 34.4% reduction in cumulative L_2 actuation cost and a 25.7% reduction in peak yaw rate, achieved solely through damping injection without altering the proportional or integral gain structure. The physical mechanism is clear; the damping term $-\mathbf{J}_{\text{vis}}^{\dagger} \kappa_D \mathbf{J}_{\text{vis}} \xi_b$ acts as velocity-dependent friction in the feature space, attenuating the aggressive forward surge that the $\ell^2/(f_v h_{\text{obj}})$ nonlinearity in the visual Jacobian produces at large distances. The trade-off is a longer depth-channel settling time (11.9 s versus 7.1 s), a trade-off that favours energy-sensitive applications over time-critical ones.

A notable aspect of the comparison is that these benefits emerge despite operating at the kinematic level ($\mathbf{M} = \mathbf{I}$), where the inertia-shaping and Coriolis-embedding terms of the full pH formulation are absent. This suggests that damping injection alone, the simplest pH-specific mechanism, already captures a practically relevant portion of the theoretical advantage. Whether the remaining energy-shaping terms yield further improvements on physical hardware remains an open question that motivates the torque-level deployment identified below.

The platform itself proved adequate as a comparative testbed: within-side repeatability was consistent across all five experimental stages, and the reduced-pH equivalence verification confirmed that observed performance differences are attributable to the pH-specific terms rather than implementation artefacts. At the same time, the experiments exposed several limitations. The left–right mechanical asymmetry, the sensitivity of the ℓ feature to partial bounding-box occlusion at high speeds, and the reliance on a general-purpose COCO-trained detector all constrain the current results to a narrow operating envelope: a single initial condition, a static target, and flat-floor operation.

The initial implementation used normalised image coordinates with $\lambda_v = 1.92$. A subsequent update converted the pipeline to pixel coordinates by scaling this value with the image width rather than the image height, yielding $f_{\text{eff}} = \lambda_v \times W = 1.92 \times 1280 = 2457.6$ px, which substantially exceeds the calibrated vertical focal length of $f_{\text{px}} = 1403.5$ px (Section V-C). As a result, the depth estimate $\hat{Z}$ is approximately $1.75\times$ too large, causing the rover to perceive the target as farther away than it actually is. This inflated $\hat{Z}$ reduces the Jacobian entries that contain $1/\hat{Z}$ (such as $\ell/\hat{Z}$), making the pseudo-inverse larger and the velocity commands more aggressive for a given gain, an effect amplified at large distances by the $\ell^2/(f_v h_{\text{obj}})$ nonlinearity in the visual Jacobian. However, because the IBVS setpoint is specified in image space ($\ell^* = 468$ px) and both controllers share the identical Jacobian throughout all experiments, the systematic scaling did not influence the relative controller comparison as the empirically tuned gains absorb the magnitude difference. The corrected pixel-space focal parameter is adopted in the updated codebase released with this thesis for use in future experiments.

Looking forward, the most impactful next step is a torque-level pH-IBVS deployment with identified motor dynamics, which would activate the full energy-shaping guarantees of Muñoz-Arias et al. [20] and allow direct monitoring of the closed-loop Hamiltonian H_d as a real-time stability certificate. On the other side, replacing the general-purpose YOLOv5 model with a single-class detector fine-tuned on the target object, adopting a 360° panoramic camera to eliminate field-of-view constraints, or a camera with a faster frame rate to increase the computation speed further. Additional directions include depth-channel gain scheduling, either conditioning $\kappa_{P,\ell}$ on the current error magnitude or reformulating the feature as $\ln(\ell/\ell^*)$ to linearise the depth mapping, and broader experimental validation with moving targets, multiple starting offsets, and external disturbances. Together, these extensions would move the platform from a proof-of-concept comparison toward a versatile testbed for energy-conditioned visual servoing in unstructured environments.

References

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

[2] A. G. S. Conceição, C. E. T. Dórea, L. Martinez, and E. R. De Pieri, "Design and implementation of model-predictive control with friction compensation on an omnidirectional mobile robot," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 467–476, 2014, Published online 2013.

[3] M. Ito, K. Tachi, K. Sugiura, Y. Suzuki, and Y. Yamazaki, "Robodragons 2026 extended team description," in *RoboCup 2026 Team Description Papers: Small Size League*, School of Information Science and Technology, Aichi Prefectural University, Aichi Prefectural University, Nagakute, Aichi, Japan, 2026.

[4] M. Nakayama and M. Ito, "Visual feedback interception of a moving ball by an omni-wheeled mobile robot with an on-board camera," 2020.

[5] Kota, "Image-based visual servoing with an omnidirectional mobile robot equipped with an on-board camera: Reducing steady-state deviations," Master's thesis, Graduate School of Information Science, Department of Media and Information Sciences, Jan. 14, 2025.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[7] G. Jocher et al., *YOLOv5 by Ultralytics*, version 7.0, 2022.

[8] Z. Machkour, D. Ortiz-Arroyo, and P. Durdevic, "Classical and deep learning based visual servoing systems: A survey on state of the art," *Journal of Intelligent & Robotic Systems*, vol. 104, p. 11, 2022.

[9] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[10] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, 1st ed. New York: John Wiley & Sons, Inc., 2005.

[11] X. Huang et al., "A survey on visual servoing for wheeled mobile robots," *International Journal of Intelligent Robotics and Applications*, vol. 5, pp. 1–27, 2021.

[12] K. Hashimoto, "A review on vision-based control of robot manipulators," *Advanced Robotics*, vol. 17, no. 10, pp. 969–991, 2003.

[13] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME Transactions on Mechatronics*, vol. 5, no. 1, pp. 39–48, 2000.

[14] J. C. Willems, "Dissipative dynamical systems Part I: General theory," *Archive for Rational Mechanics and Analysis*, vol. 45, no. 5, pp. 321–351, 1972.

[15] B. M. Maschke and A. J. van der Schaft, "Port-controlled Hamiltonian systems: Modelling origins and systemtheoretic properties," in *Proc. 2nd IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, Bordeaux, France, 1992, pp. 282–288.

[16] A. van der Schaft, "Port-Hamiltonian modeling for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 393–416, 2020.

[17] R. Ortega, A. J. van der Schaft, B. Maschke, and G. Escobar, "Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems," *Automatica*, vol. 38, no. 4, pp. 585–596, 2002.

[18] M. Fujita, H. Kawai, and M. W. Spong, "Passivity-based dynamic visual feedback control for three-dimensional target tracking: Stability and L_2-gain performance analysis," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 1, pp. 40–52, 2007.

[19] D. A. Dirksz, J. M. A. Scherpen, and M. Steinbuch, "A Port-Hamiltonian Approach to Visual Servo Control of a Pick and Place System," *Asian Journal of Control*, vol. 16, no. 3, pp. 703–713, May 2014.

[20] M. Muñoz-Arias, M. Ito, and J. M. A. Scherpen, "A port-hamiltonian approach to visual servo control for standard mechanical systems," Preprint submitted to *Automatica*, Mar. 2025.

[21] H. Ma, M. Munoz-Arias, J. M. A. Scherpen, and A. Macchelli, "An energy-based approach to the force-impedance control problem for robot manipulators," *IEEE Transactions on Control Systems Technology*, vol. 34, no. 1, Jan. 2026.

[22] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, 1st. Cambridge University Press, 2017, Preprint accessed: 2026-02-28.

[23] Sabertooth, *Rover 5 introduction manual*, Accessed: 2026-02-28, Mouser Electronics, n.d.

IX. Appendix

A. Image Jacobian $\mathbf{J}_f$

This subsection derives the two-row image Jacobian $\mathbf{J}_f \in \mathbb{R}^{2\times 6}$ that maps the camera spatial velocity $\xi_c = [\dot{c}_x^c,\, \dot{c}_y^c,\, \dot{c}_z^c,\, \omega_x^c,\, \omega_y^c,\, \omega_z^c]^\top$ to the feature rates $\dot{\mathbf{s}} = [\dot{\mu},\, \dot{\ell}]^\top$, as introduced in (8).

a) Step 1 — Rigid-body point velocity.: For a *static* target observed by a moving camera, the time derivative of the target position expressed in the camera frame is [10]:

$$ {}^c\dot{\mathbf{p}} = -{}^c\mathbf{v}_c - [\omega_c]_\times\, {}^c\mathbf{p}, \qquad [\omega_c]_\times = \begin{bmatrix} 0 & -\omega_z^c & \omega_y^c \\ \omega_z^c & 0 & -\omega_x^c \\ -\omega_y^c & \omega_x^c & 0 \end{bmatrix}. \tag{19} $$

b) Step 2 — Planar reduction.: Under planar motion the camera height is constant (${}^cy \approx 0$) and only the yaw component of the angular velocity is non-zero ($\omega_x^c = \omega_z^c = 0$). Expanding (19) under these constraints gives the two scalar rates required for the feature derivatives:

$$ {}^c\dot{x} = -\dot{c}_x^c - \omega_y^c\, {}^cz, \qquad {}^c\dot{z} = -\dot{c}_z^c + \omega_y^c\, {}^cx. \tag{20} $$

c) Step 3 — Differentiation of μ (row 1).: From the pinhole projection (4), $\mu = f_\mu\, {}^cx/Z$ with $Z = {}^cz$. Applying the quotient rule and substituting (20):

$$ \begin{aligned} \dot{\mu} &= \frac{f_\mu}{Z}\, {}^c\dot{x} - \frac{\mu}{Z}\, {}^c\dot{z} \\ &= \frac{f_\mu}{Z}\left(-\dot{c}_x^c - \omega_y^c Z\right) - \frac{\mu}{Z}\left(-\dot{c}_z^c + \omega_y^c\, {}^cx\right) \\ &= -\frac{f_\mu}{Z}\,\dot{c}_x^c + \frac{\mu}{Z}\,\dot{c}_z^c - f_\mu\,\omega_y^c - \frac{\mu\, {}^cx}{Z}\,\omega_y^c. \end{aligned} \tag{21} $$

Replacing the last term via ${}^cx = \mu Z/f_\mu$ (from (4)) yields the compact form used in (8):

$$ \boxed{\dot{\mu} = -\frac{f_\mu}{\hat{Z}}\,\dot{c}_x^c + \frac{\mu}{\hat{Z}}\,\dot{c}_z^c - \left(f_\mu + \frac{\mu^2}{f_\mu}\right)\omega_y^c.} \tag{22} $$

Only f_μ appears, because μ is defined through the *horizontal* focal parameter.

d) Step 4 — Differentiation of ℓ (row 2).: From the apparent-size model (6), $\ell = f_\nu\, h_{\text{obj}}/Z$:

$$ \dot{\ell} = -\frac{\ell}{Z}\,\dot{Z}. \tag{23} $$

The depth rate follows from (20): $\dot{Z} = {}^c\dot{z} = -\dot{c}_z^c + \omega_y^c\, {}^cx$. Substituting and replacing ${}^cx = \mu Z/f_\mu$:

$$ \boxed{\dot{\ell} = \frac{\ell}{\hat{Z}}\,\dot{c}_z^c - \frac{\ell\,\mu}{f_\mu}\,\omega_y^c.} \tag{24} $$

Note that f_ν does *not* appear explicitly: it is absorbed into the measured value ℓ. The coupling with μ enters through the horizontal relation ${}^cx = \mu\hat{Z}/f_\mu$.

e) Step 5 — Matrix assembly.: Writing (22) and (24) in matrix form with $\xi_c = [\dot{c}_x^c,\, \dot{c}_y^c,\, \dot{c}_z^c,\, \omega_x^c,\, \omega_y^c,\, \omega_z^c]^\top$ gives the image Jacobian stated in (8):

$$ \boxed{\mathbf{J}_f(\mu,\ell,\hat{Z}) = \begin{bmatrix} -\dfrac{f_\mu}{\hat{Z}} & 0 & \dfrac{\mu}{\hat{Z}} & 0 & -\left(f_\mu + \dfrac{\mu^2}{f_\mu}\right) & 0 \\ 0 & 0 & \dfrac{\ell}{\hat{Z}} & 0 & -\dfrac{\ell\,\mu}{f_\mu} & 0 \end{bmatrix}} \tag{25} $$

Non-zero columns: 1 ($\dot{c}_x^c$), 3 ($\dot{c}_z^c$), 5 (ω_y^c). Columns 2, 4, 6 vanish identically under the planar constraints of Step 2.

B. Geometric Coupling $\mathbf{J}_c$

This subsection derives the constant 6×3 matrix $\mathbf{J}_c$ that maps the rover body twist $\xi_b = [v_x,\, v_y,\, \omega_z]^\top$ to the camera spatial velocity ξ_c, as stated in (9).

a) Step 1 — Rotation matrices.: The world-to-robot rotation uses ZYZ Euler angles (ϕ,θ,ψ) [3]:

$$ \mathbf{R}_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}. \tag{26} $$

The Euler-angle rate to angular velocity transformation [5]:

$$ \mathbf{T}_\omega(\phi,\theta) = \begin{bmatrix} 0 & -\sin\phi & \cos\phi\,\sin\theta \\ 0 & \cos\phi & \sin\phi\,\sin\theta \\ 1 & 0 & \cos\theta \end{bmatrix}. \tag{27} $$

b) Step 2 — General kinematic chain.: The camera spatial velocity is related to the generalised coordinate rates $\dot{\mathbf{q}}$ through [3]:

$$ \begin{bmatrix} {}^c\dot{\mathbf{p}}_c \\ {}^c\boldsymbol{\omega}_c \end{bmatrix} = \underbrace{\begin{bmatrix} {}^w\mathbf{R}_c^\top & \mathbf{0} \\ \mathbf{0} & {}^w\mathbf{R}_c^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_\omega \end{bmatrix} \begin{bmatrix} \partial\, {}^w\mathbf{p}_c/\partial\mathbf{q} \\ \partial\, {}^w\boldsymbol{\alpha}_c/\partial\mathbf{q} \end{bmatrix}}_{\mathbf{J}_c(\phi)} \underbrace{\mathbf{R}_z(\phi)\,\xi_b}_{\dot{\mathbf{q}}}. \tag{28} $$

c) Step 3 — Forward kinematics.: The camera is rigidly mounted at longitudinal offset d_c forward of the rover centre and at height h_c (Figure 3):

$$ {}^w\mathbf{p}_c = \begin{bmatrix} x + d_c\cos\phi \\ y + d_c\sin\phi \\ h_c \end{bmatrix}, \tag{29} $$

with partial derivatives:

$$ \frac{\partial\, {}^w\mathbf{p}_c}{\partial\mathbf{q}} = \begin{bmatrix} 1 & 0 & -d_c\sin\phi \\ 0 & 1 & d_c\cos\phi \\ 0 & 0 & 0 \end{bmatrix}, \qquad \frac{\partial\, {}^w\boldsymbol{\alpha}_c}{\partial\mathbf{q}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{30} $$

d) Step 4 — Planar simplification ($\theta = \psi = 0$).: With no pitch or roll: $\mathbf{T}_\omega \to \mathbf{I}_3$ and ${}^w\mathbf{R}_c^\top = \mathbf{R}_z(-\phi)$. The product $\mathbf{R}_z(-\phi)\,\mathbf{R}_z(\phi) = \mathbf{I}_3$ cancels all ϕ-dependence, yielding the

constant geometric Jacobian stated in (9):

$$\mathbf{J}_c = \begin{bmatrix} 0 & -1 & -d_c \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{6\times 3} \rightarrow \begin{aligned} \dot{c}_x^c &= -v_y - d_c\,\omega_z, \\ \dot{c}_z^c &= v_x, \\ \omega_y^c &= -\omega_z. \end{aligned} \tag{31}$$

The three non-trivial rows encode: (i) lateral camera motion from the rover's sideways velocity and the yaw lever-arm through d_c; (ii) the camera depth rate equalling the rover's forward velocity; and (iii) rover yaw mapping directly to camera pan with a sign reversal due to the frame-axis convention (Figure 3).

C. Visual Jacobian $\mathbf{J}_{\text{vis}} = \mathbf{J}_f \mathbf{J}_c$

With $\mathbf{J}_f$ (25) and $\mathbf{J}_c$ (31) in hand, we compute the 2×3 visual interaction matrix by substituting the geometric relations from (31) into the feature rate expressions (22) and (24).

a) Column 1 (v_x).: From (31): only $\dot{c}_z^c = v_x$ is non-zero. Substituting into (22) and (24):

$$J_{\text{vis},11} = \frac{\mu}{\hat{Z}}\cdot 1 = \frac{\mu}{\hat{Z}}, \qquad J_{\text{vis},21} = \frac{\ell}{\hat{Z}}\cdot 1 = \frac{\ell}{\hat{Z}}.$$

b) Column 2 (v_y).: From (31): only $\dot{c}_x^c = -v_y$ contributes.

$$J_{\text{vis},12} = \left(-\frac{f_\mu}{\hat{Z}}\right)(-1) = \frac{f_\mu}{\hat{Z}}, \qquad J_{\text{vis},22} = 0\cdot(-1) = 0.$$

The exact zero in row 2 confirms the structural v_y–ℓ decoupling: lateral motion does not affect the apparent size.

c) Column 3 (ω_z).: Both $\dot{c}_x^c = -d_c\,\omega_z$ and $\omega_y^c = -\omega_z$ contribute:

$$J_{\text{vis},13} = \left(-\frac{f_\mu}{\hat{Z}}\right)(-d_c) + \left(-f_\mu - \frac{\mu^2}{f_\mu}\right)(-1) = \frac{f_\mu\, d_c}{\hat{Z}} + f_\mu + \frac{\mu^2}{f_\mu}, \tag{32}$$

$$J_{\text{vis},23} = 0\cdot(-d_c) + \left(-\frac{\ell\,\mu}{f_\mu}\right)(-1) = \frac{\ell\,\mu}{f_\mu}. \tag{33}$$

d) Result — compact form.: Assembling all six entries gives the visual interaction matrix stated in (10):

$$\mathbf{J}_{\text{vis}}(\mu,\ell,\hat{Z}) = \begin{bmatrix} \dfrac{\mu}{\hat{Z}} & \dfrac{f_\mu}{\hat{Z}} & f_\mu + \dfrac{\mu^2}{f_\mu} + \dfrac{f_\mu\, d_c}{\hat{Z}} \\ \dfrac{\ell}{\hat{Z}} & 0 & \dfrac{\ell\,\mu}{f_\mu} \end{bmatrix} \tag{34}$$

where the depth $\hat{Z}$ is not measured directly but computed from the apparent height via (6).

e) Depth-substituted form.: Replacing $\hat{Z} = f_v\, h_{\text{obj}}/\ell$ from (6) (equivalently $1/\hat{Z} = \ell/(f_v\, h_{\text{obj}})$) eliminates the explicit depth dependence:

$$\mathbf{J}_{\text{vis}}(\mu,\ell) = \begin{bmatrix} \dfrac{\mu\,\ell}{f_v\, h_{\text{obj}}} & \dfrac{f_\mu\,\ell}{f_v\, h_{\text{obj}}} & f_\mu + \dfrac{\mu^2}{f_\mu} + \dfrac{f_\mu\, d_c\,\ell}{f_v\, h_{\text{obj}}} \\ \dfrac{\ell^2}{f_v\, h_{\text{obj}}} & 0 & \dfrac{\ell\,\mu}{f_\mu} \end{bmatrix} \tag{35}$$

In this form both focal parameters appear: f_μ from $\mathbf{J}_f$ (horizontal image geometry) and f_v from the depth model (6). Two structural properties are immediate:

- $(2,2)$ **zero** — exact and depth-independent, confirming that lateral velocity v_y does not affect the apparent size ℓ. This decoupling is a consequence of the holonomic mecanum drive and the forward-facing camera arrangement.
- $(2,1)$ **entry** $\propto \ell^2$ — the quadratic nonlinearity that amplifies the forward velocity command v_x when the target is far away (ℓ small), producing the aggressive forward surge discussed in Section VIII.

D. KiCad Drawing

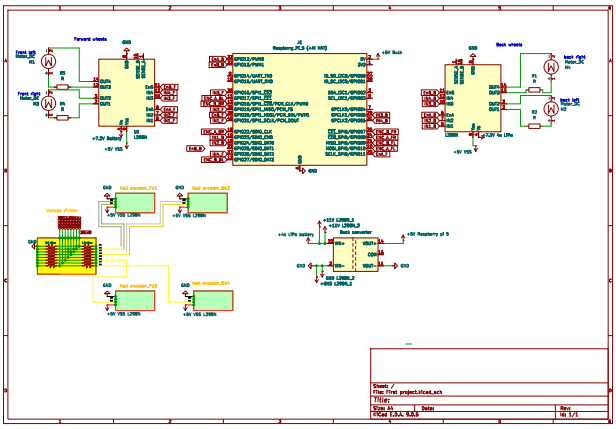

Fig. 19: System integration schematic in KiCad illustrating the interface between the Raspberry Pi 5, Hailo-8L accelerator, L298N motor drivers and powerdistribution. The 4S LiPo battery powers the motor drivers directly while the computational unit is regulated via a Pololu buck converter. From the VCC of the driver modules, voltage shifters convert quadrature encoder signals from 5 V to 3.3 V for Raspberry Pi GPIO compatibility. The rover's wheels are driven by DC motors regulated by the L298N H-bridges, with quadrature encoders mounted on each motor shaft providing velocity feedback at the wheel level. Detailed component specifications and GPIO assignments are provided in Table I.

TABLE IV: Time-domain metrics for holonomic (v_y-enabled = 1) left/right repeatability trials (mean $\pm$ std over 3 runs). T_r: rise time, T_s: settling time, SSE: steady-state error.

	Left (3 runs)	**Right (3 runs)**
Duration [s]	30	30
$K_{P,\mu}$ (proportional)	10.0	10.0
$K_{P,\ell}$ (proportional)	0.4	0.4
Lateral alignment (μ)		
$T_r(\mu)$ [s]	2.564 ± 0.032	3.897 ± 0.113
$T_s(\mu)$ [s]	12.223 ± 1.274	4.8721 ± 1.074
SSE(μ) [px]	5.569 ± 1.079	2.744 ± 0.201
Depth regulation (ℓ)		
$T_r(\ell)$ [s]	8.272 ± 0.114	7.053 ± 0.576
$T_s(\ell)$ [s]	18.414 ± 2.123	13.066 ± 1.551
SSE(ℓ) [px]	2.444 ± 0.686	1.376 ± 0.169

TABLE V: Holonomic P-SH gain sets and corresponding time-domain metrics for lateral alignment μ and depth regulation ℓ.

	gain1	**gain2**	**gain3**	**gain4**	**gain5**	**gain6**
Controller gain configuration						
$K_{P,\mu}$ (prop.)	1.5	8.0	17.0	20.0	30.0	30.0
$K_{P,\ell}$ (prop.)	0.2	0.3	0.5	0.5	0.6	0.2
$K_{I,\mu}$ (int.)	0.0	0.0	0.0	0.0	0.0	0.0
$K_{I,\ell}$ (int.)	0.0	0.0	0.0	0.0	0.0	0.0
Lateral alignment (μ)						
$T_r(\mu)$ [s]	9.333	4.100	3.991	3.594	3.671	2.652
$T_s(\mu)$ [s]	16.180	6.652	4.731	4.196	4.881	3.925
SSE(μ) [px]	8.884	3.065	3.317	3.311	3.382	3.893
Depth regulation (ℓ)						
$T_r(\ell)$ [s]	—	8.903	5.137	6.076	4.578	19.570
$T_s(\ell)$ [s]	—	16.993	8.090	14.401	7.329	—
SSE(ℓ) [px]	—	2.328	1.914	2.376	2.244	—

TABLE VI: Proportional-Integral ($v_y = 1$) gain sets and corresponding time-domain metrics for lateral alignment μ and depth regulation ℓ.

	gain3	**gain3_1**	**gain3_2**	**gain3_3**	**gain3_4**	**gain3_5**
Controller gain configuration						
$K_{P,\mu}$ (prop.)	17.0	17.0	17.0	17.0	17.0	17.0
$K_{P,\ell}$ (prop.)	0.5	0.5	0.5	0.5	0.5	0.5
$K_{I,\mu}$ (int.)	0.0	0.003	0.005	0.0	0.0	0.002
$K_{I,\ell}$ (int.)	0.0	0.0	0.0	0.003	0.005	0.001
Lateral alignment (μ)						
$T_r(\mu)$ [s]	3.991	3.640	3.802	3.760	3.729	3.857
$T_s(\mu)$ [s]	4.713	6.019	4.343	4.028	4.296	4.361
SSE(μ) [px]	3.317	2.664	3.428	4.570	3.269	3.606
Depth regulation (ℓ)						
$T_r(\ell)$ [s]	5.137	6.560	6.889	11.213	5.637	7.081
$T_s(\ell)$ [s]	8.090	13.673	14.042	18.297	7.785	10.372
SSE(ℓ) [px]	1.914	2.511	3.023	4.152	6.900	1.936

TABLE VII: Comparative time-domain metrics for SH PI-IBVS and reduced pH controllers. Mean ± std over 3 runs per condition. Proportional gains matched; integral, damping, and leaky terms disabled ($\mathbf{K}_I = \mathbf{K}_D = \lambda_{\text{leak}} = \mathbf{0}$).

Metric	**SH_P_gain3_2**	**redPH_P_right1**
Controller gain configuration		
$K_{P,\mu}$ (proportional)	17.0	17.0
$K_{P,\ell}$ (proportional)	0.5	0.5
$K_{I,\mu}$ (integral)	0.0	0.0
$K_{I,\ell}$ (integral)	0.0	0.0
$K_{D,\mu}$ (damping)	—	0.0
$K_{D,\ell}$ (damping)	—	0.0
λ_{leak}	—	0.0
Lateral alignment (μ)		
$T_r(\mu)$ [s]	3.413 ± 0.064	3.579 ± 0.176
$T_s(\mu)$ [s]	4.421 ± 0.110	4.015 ± 0.273
SSE(μ) [px]	3.588 ± 0.376	3.937 ± 0.143
Depth regulation (ℓ)		
$T_r(\ell)$ [s]	4.688 ± 0.389	4.744 ± 0.874
$T_s(\ell)$ [s]	7.149 ± 0.457	6.768 ± 1.097
SSE(ℓ) [px]	2.722 ± 0.677	2.435 ± 0.296
Velocities / actuation cost (FK from encoders)		
Max $\lvert\omega_z\rvert$ actual [rad/s]	2.622 ± 0.132	2.487 ± 0.277
Max $\lvert v_x\rvert$ actual [m/s]	0.765 ± 0.026	0.743 ± 0.029
$E = \int \lVert\xi_b\rVert^2 \, dt$ [(m/s)^2·s]	6.271 ± 0.581	5.591 ± 0.904

TABLE VIII: Damping gain sweep for the pH controller. Baseline SH PI-controller (redSH_PI) shown for reference. All pH variants use $K_{P,\mu} = 17.0$, $K_{P,\ell} = 0.5$, $K_{I,\mu} = 0.002$, $K_{I,\ell} = 0.001$, and $\lambda_{\text{leak}} = 0.03$.

	redSH_PI	**PIDL_gain1**	**PIDL_gain2**	**PIDL_gain3**	**PIDL_gain4**
Controller gain configuration					
$K_{P,\mu}$ (prop.)	17.0	17.0	17.0	17.0	17.0
$K_{P,\ell}$ (prop.)	0.5	0.5	0.5	0.5	0.5
$K_{I,\mu}$ (int.)	0.0	0.002	0.002	0.002	0.002
$K_{I,\ell}$ (int.)	0.0	0.001	0.001	0.001	0.001
$K_{D,\mu}$ (damp.)	0.0	0.1	0.3	0.5	0.5
$K_{D,\ell}$ (damp.)	0.0	0.0	0.0	0.1	0.3
λ_{leak}	0.0	0.03	0.03	0.03	0.03
Lateral alignment (μ)					
$T_r(\mu)$ [s]	3.360	3.230	3.555	3.518	3.718
$T_s(\mu)$ [s]	3.898	5.512	8.859	6.538	3.887
SS-MAE(μ) [px]	3.846	4.200	8.855	2.377	4.626
Depth regulation (ℓ)					
$T_r(\ell)$ [s]	4.942	5.512	5.000	6.808	7.312
$T_s(\ell)$ [s]	6.281	9.172	7.516	15.006	13.186
SS-MAE(ℓ) [px]	2.207	1.570	3.186	1.979	1.793
Velocities / actuation cost (FK from encoders)					
Max $\lvert\omega_z\rvert$ [rad/s]	2.877	2.487	2.337	1.871	2.721
Max $\lvert v_x\rvert$ [m/s]	0.770	0.772	0.799	0.757	0.770
Max $\lvert v_y\rvert$ [m/s]	0.463	0.448	0.462	0.377	0.770
E [(m/s)^2·s]	6.551	5.402	4.310	3.382	4.696

TABLE IX: Final comparative performance metrics: SH PI-IBVS (*SH_P_gain3_2*) versus pH-IBVS with damping and leaky integrator (*redPH_PIDL_gain3*). Mean ± std over 3 runs.

Parameter / Metric	**redPH_PIDL_gain3**	**SH_P_gain3_2**
Controller gain configuration		
$K_{P,\mu}$ (proportional)	17.0	17.0
$K_{P,\ell}$ (proportional)	0.5	0.5
$K_{I,\mu}$ (integral)	0.002	0.0
$K_{I,\ell}$ (integral)	0.001	0.0
$K_{D,\mu}$ (damping)	0.5	—
$K_{D,\ell}$ (damping)	0.1	—
λ_{leak}	0.03	—
Lateral alignment (μ)		
$T_r(\mu)$ [s]	3.715 ± 0.173	3.413 ± 0.064
$T_s(\mu)$ [s]	4.006 ± 0.313	4.421 ± 0.110
SS-MAE(μ) [px] [$t \geq T_s$]	3.256 ± 0.669	3.588 ± 0.376
Depth regulation (ℓ)		
$T_r(\ell)$ [s]	6.086 ± 0.603	4.688 ± 0.389
$T_s(\ell)$ [s]	11.931 ± 2.185	7.149 ± 0.457
SS-MAE(ℓ) [px] [$t \geq T_s$]	1.713 ± 0.118	2.722 ± 0.677
Velocities / actuation cost (FK from encoders)		
Max $\lvert\omega_z\rvert$ actual [rad/s]	1.949 ± 0.109	2.622 ± 0.132
Max $\lvert v_x\rvert$ actual [m/s]	0.764 ± 0.014	0.765 ± 0.026
$E = \int \lVert\xi_b\rVert^2 \, \mathrm{d}t$ [(m/s)^2·s]	4.117 ± 0.512	6.271 ± 0.581

Algorithm 1 Vision Pipeline and Feature Extraction (30 Hz)

Require: GStreamer pipeline initialised, Hailo-8L loaded with YOLOv5 model
Ensure: Visual features $\mathbf{s}(t) = [\mu(t), \ell(t)]^\top$ and depth estimate $\hat{Z}(t)$

Constants: $W = 1280$, $H = 720$, $f_\mu = 1403$ px, $h_{\text{obj}} = 0.23$ m, $\theta_{\text{conf}} = 0.4$
Initialise: $\mathbf{s} \leftarrow [\text{NaN}, \text{NaN}]^\top$, $\hat{Z} \leftarrow \text{NaN}$, $n_{\text{stable}} \leftarrow 0$
while pipeline active **do**
    frame $\leftarrow$ AcquireFrame() ▷ 30 fps USB camera
    detections $\leftarrow$ HailoInference(frame) ▷ YOLOv5 on Hailo-8L NPU
    detected $\leftarrow$ False
    for each detection d in detections **do**
        if d.label = "bottle" **and** d.conf $> \theta_{\text{conf}}$ **then**
            bbox $\leftarrow d$.bbox ▷ $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ normalised to $[0,1]$
            $\mu \leftarrow \frac{1}{2}(x_{\min} + x_{\max}) \cdot W - \frac{W}{2}$ ▷ Horizontal offset from image centre [px]
            $f_v \leftarrow \frac{1}{2}(y_{\min} + y_{\max}) \cdot H - \frac{H}{2}$ ▷ Vertical offset from image centre [px]
            $\ell \leftarrow (y_{\max} - y_{\min}) \cdot H$ ▷ Apparent bounding-box height [px]
            $\hat{Z} \leftarrow f_v \cdot H \cdot h_{\text{obj}} \,/\, \ell$ ▷ Monocular depth estimate (Eq. 6)
            detected $\leftarrow$ True; **break**
        end if
    end for
    if detected **then** ▷ Hysteresis stability filter
        $n_{\text{stable}} \leftarrow \min(n_{\text{stable}} + 1,\ 5)$
    else
        $n_{\text{stable}} \leftarrow \max(n_{\text{stable}} - 1,\ 0)$
    end if
    if $n_{\text{stable}} \geq 2$ **then**
        $\mathbf{s} \leftarrow [\mu, \ell]^\top$ ▷ Stable detection — publish features
    else
        $\mathbf{s} \leftarrow [\text{NaN}, \text{NaN}]^\top$ ▷ Insufficient confidence — suppress output
    end if
    PublishFeatures($\mathbf{s}, \hat{Z}, f_v$) ▷ Send to control layer
end while

Algorithm 2 SH PI-IBVS Control Law (Eq. 11)

Require: Visual features $\mathbf{s} = [\mu, \ell]^\top$ [px], depth estimate $\hat{Z}$, vertical offset f_v from Algorithm 1
Ensure: Commanded body twist $\xi_b = [v_x, v_y, \omega_z]^\top$ and wheel RPM targets ω^*

Constants: $\mathbf{s}^* = [\mu^*, \ell^*]^\top$, $d_c = 0.09\,\text{m}$, $r = 0.06\,\text{m}$, $\ell_x = 0.093\,\text{m}$, $\ell_y = 0.087\,\text{m}$
Gains: $\kappa_P = \text{diag}(\kappa_{P,\mu}, \kappa_{P,\ell}) > 0$, $\kappa_I = \text{diag}(\kappa_{I,\mu}, \kappa_{I,\ell}) \geq 0$
Initialise: $\xi_{\text{int}} \leftarrow \mathbf{0}$, $t_{\text{prev}} \leftarrow 0$
while control active at 30 Hz **do**
  $\Delta t \leftarrow t_{\text{now}} - t_{\text{prev}}$; $t_{\text{prev}} \leftarrow t_{\text{now}}$
  if $\mathbf{s}$ is valid **then** ▷ Stable detection from Algorithm 1
    $\mathbf{e} \leftarrow \mathbf{s} - \mathbf{s}^*$ ▷ Feature error vector [px]
    $\xi_{\text{int}} \leftarrow \xi_{\text{int}} + \mathbf{e}\,\Delta t$ ▷ Discrete integral accumulation
    $\xi_{\text{int}} \leftarrow \textsc{Clamp}(\xi_{\text{int}}, -\xi_{\max}, +\xi_{\max})$ ▷ Hard-clamp anti-windup
    $\mathbf{J}_f \leftarrow \textsc{ImageJacobian}(\mu, \ell, \hat{Z}, f_v)$ ▷ $\in \mathbb{R}^{2\times 6}$, Eq. 8
    $\mathbf{J}_c \leftarrow \textsc{GeometricJacobian}(d_c)$ ▷ $\in \mathbb{R}^{6\times 3}$, Eq. 9
    $\mathbf{J}_{\text{vis}} \leftarrow \mathbf{J}_f \mathbf{J}_c$ ▷ Composed visual Jacobian $\in \mathbb{R}^{2\times 3}$
    $\mathbf{J}_{\text{vis}}^\dagger \leftarrow (\mathbf{J}_{\text{vis}}^\top \mathbf{J}_{\text{vis}})^{-1} \mathbf{J}_{\text{vis}}^\top$ ▷ Moore–Penrose right pseudo-inverse
    $\mathbf{u} \leftarrow \kappa_P \mathbf{e} + \kappa_I \xi_{\text{int}}$ ▷ PI control signal
    $\xi_b \leftarrow -\mathbf{J}_{\text{vis}}^\dagger \mathbf{u}$ ▷ Commanded body twist (Eq. 11)
    $\xi_b \leftarrow \textsc{Saturate}(\xi_b, v_{\max}, \omega_{\max})$ ▷ Velocity limits
  else
    $\xi_b \leftarrow \mathbf{0}$; $\xi_{\text{int}} \leftarrow \mathbf{0}$ ▷ No detection — stop motors and reset integrator
  end if
  $\omega^* \leftarrow \textsc{TwistToWheelRPM}(\xi_b)$ ▷ Mecanum inverse kinematics (Eq. 16)
  $\omega^* \leftarrow \textsc{Clamp}(\omega^*, -140, 140)$ ▷ Per-wheel RPM saturation
  $\textsc{PublishRPMTargets}(\omega^*)$ ▷ Send to inner-loop thread (Algorithm 4)
end while

Algorithm 3 pH-IBVS Control Law with Implicit Damping and Leaky Integration (Eq. 14)

Require: Visual features $\mathbf{s} = [\mu, \ell]^\top$ [px], depth estimate $\hat{Z}$, vertical offset f_v from Algorithm 1
Ensure: Commanded body twist $\xi_b = [v_x, v_y, \omega_z]^\top$

Gains: $\kappa_P = \mathrm{diag}(\kappa_{P,\mu}, \kappa_{P,\ell})$, $\kappa_I = \mathrm{diag}(\kappa_{I,\mu}, \kappa_{I,\ell})$, $\kappa_D = \mathrm{diag}(\kappa_{D,\mu}, \kappa_{D,\ell})$, λ_{leak}, $K_{\text{aw}} = 0$, $\lambda_{\text{dls}} = 0$
Initialise: $\mathbf{z} \leftarrow \mathbf{0}$ (leaky integral state), $H_{\text{prev}} \leftarrow 0$ (Hamiltonian monitor)
while control active at 30 Hz **do**
  if $\mathbf{s}$ is valid **then**
    Step 1 — Feature error: $\mathbf{e}_\sigma \leftarrow \mathbf{s} - \mathbf{s}^*$ ▷ [px]
    Step 2 — Visual Jacobian: $\mathbf{J}_{\text{vis}} \leftarrow \mathbf{J}_f \mathbf{J}_c \in \mathbb{R}^{2\times 3}$
      $\mathbf{J}^\dagger \leftarrow \mathbf{J}_{\text{vis}}^\top \left(\mathbf{J}_{\text{vis}} \mathbf{J}_{\text{vis}}^\top + \lambda_{\text{dls}}^2 \mathbf{I}_2\right)^{-1}$ ▷ Damped least-squares pseudo-inverse $\in \mathbb{R}^{3\times 2}$
    Step 3 — Leaky integrator: $\dot{\mathbf{z}} \leftarrow \mathbf{e}_\sigma - \lambda_{\text{leak}} \mathbf{z}$; $\mathbf{z} \leftarrow \mathbf{z} + \Delta t\, \dot{\mathbf{z}}$ ▷ Structural anti-windup
    Step 4 — Control law (implicit damping):
      $\nabla H \leftarrow \kappa_P \mathbf{e}_\sigma + \kappa_I \mathbf{z}$ ▷ Closed-loop energy gradient
      $\mathbf{b} \leftarrow -\mathbf{J}^\dagger \nabla H$ ▷ Undamped velocity command
      $\mathbf{A} \leftarrow \mathbf{I}_3 + \mathbf{J}^\dagger \kappa_D \mathbf{J}_{\text{vis}}$ ▷ Implicit damping matrix $\in \mathbb{R}^{3\times 3}$
      $\xi_b^{\text{unsat}} \leftarrow \mathbf{A}^{-1} \mathbf{b}$ ▷ Solve $(\mathbf{I} + \mathbf{J}^\dagger \kappa_D \mathbf{J}_{\text{vis}})\, \xi_b = \mathbf{b}$
      $\xi_b \leftarrow \textsc{Saturate}(\xi_b^{\text{unsat}}, v_{\max}, \omega_{\max})$ ▷ Velocity limits
    Step 5 — Back-calculation anti-windup:
      $\Delta\mathbf{v} \leftarrow \xi_b - \xi_b^{\text{unsat}}$ ▷ Saturation residual (zero when unsaturated)
      $\mathbf{z} \leftarrow \mathbf{z} + K_{\text{aw}} \mathbf{J}_{\text{vis}} \Delta\mathbf{v}$ ▷ Project residual to feature space, discharge integrator
      $\mathbf{z} \leftarrow \textsc{Clamp}(\mathbf{z}, -\mathbf{z}_{\max}, +\mathbf{z}_{\max})$ ▷ Hard-clamp safety backstop
    Step 6 — Hamiltonian monitor:
      $H_d \leftarrow \frac{1}{2} \mathbf{e}_\sigma^\top \kappa_P \mathbf{e}_\sigma + \frac{1}{2} \mathbf{z}^\top \kappa_I^{-1} \mathbf{z}$ ▷ $H_d \geq 0$; $\Delta H_d \leq 0$ is passivity certificate
      $\Delta H \leftarrow H_d - H_{\text{prev}}$; $H_{\text{prev}} \leftarrow H_d$
  else
    $\xi_b \leftarrow \mathbf{0}$; $\mathbf{z} \leftarrow \mathbf{0}$; $H_{\text{prev}} \leftarrow 0$ ▷ No detection — stop and reset all states
  end if
  $\textsc{PublishRPMTargets}\big(\textsc{TwistToWheelRPM}(\xi_b)\big)$ ▷ Forward to inner loop (Algorithm 4)
end while

Algorithm 4 Inner-Loop Wheel Velocity Control (50 Hz Daemon Thread)

Require: RPM targets $\omega^* = [\omega_1^*, \omega_2^*, \omega_3^*, \omega_4^*]^\top$ from Algorithm 2 or 3
Ensure: PWM duty cycles $\mathbf{D} = [D_1, D_2, D_3, D_4]^\top$ applied to L298N motor drivers

Constants: CPR $= 333$ counts/rev, $\Delta t = 0.02$ s (50 Hz), $D_{\max} = 100\%$, PWM frequency $= 1$ kHz
Per-wheel gains: $K_{p,i} = 0.6$, $K_{i,i} = 0.02$, $I_{\max}$ (maximum integral duty contribution)
Initialise: $\mathbf{D} \leftarrow \mathbf{0}$, $\eta \leftarrow \mathbf{0}$ (integral states), $\text{pos}_i^{\text{prev}} \leftarrow \textsc{ReadEncoder}(i)$ for $i = 1 \ldots 4$
while control active **do**
    $t_{\text{start}} \leftarrow \textsc{GetTime}()$
    for $i = 1$ **to** 4 **do** ▷ Read all four quadrature encoders
        $\text{pos}_i \leftarrow \textsc{ReadEncoder}(i)$
        $\Delta\text{pos}_i \leftarrow \text{pos}_i - \text{pos}_i^{\text{prev}}$; $\text{pos}_i^{\text{prev}} \leftarrow \text{pos}_i$
        $\hat{\omega}_i \leftarrow \dfrac{\Delta\text{pos}_i}{\Delta t} \cdot \dfrac{60}{\text{CPR}}$ ▷ Convert encoder counts to measured RPM (Eq. 18)
    end for
    $\omega^* \leftarrow \textsc{ReadSharedTargets}()$ ▷ Thread-safe mutex-protected read from control layer
    for $i = 1$ **to** 4 **do** ▷ Positional PI controller per wheel
        if $|\omega_i^*| < 0.001$ **then** ▷ Target is zero — coast
            $D_i \leftarrow 0$; $\eta_i \leftarrow 0$ ▷ Set duty to zero and reset integrator to prevent windup carry-over
        else
            $e_i \leftarrow |\omega_i^*| - |\hat{\omega}_i|$ ▷ Speed error (magnitude)
            $\eta_i \leftarrow \eta_i + e_i \Delta t$ ▷ Accumulate integral
            $\eta_i \leftarrow \textsc{Clamp}\big(\eta_i,\ -I_{\max}/K_{i,i},\ +I_{\max}/K_{i,i}\big)$ ▷ Anti-windup: bound integral contribution to $\pm I_{\max}$ %
            $D_i \leftarrow \textsc{Clamp}\big(K_{p,i}\, e_i + K_{i,i}\, \eta_i,\ 0,\ D_{\max}\big)$ ▷ Positional PI output, clamped to $[0, 100]$ %
        end if
        $\textsc{SetDirection}(i, \text{sign}(\omega_i^*))$ ▷ Set H-bridge polarity via GPIO
        $\textsc{SetPWM}(i, D_i)$ ▷ Apply duty cycle to L298N enable pin
    end for
    $\textsc{Sleep}\big(\max(0, \Delta t - (\textsc{GetTime}() - t_{\text{start}}))\big)$ ▷ Maintain fixed 50 Hz loop rate
end while