Fuzzy Logic Control for Fast-Tracking Gimbals in Velocity Space

Ahmed Aboabdallah[1,*], Kareem Aboelmhasen[1], Ammar Abdelwahab[1], Ahmed Gaballa[1], Heba Abokhalil[1], Abdelrahman Alamrawy[1], Hamza Frehy[1], Hanem Gebreel[1], Mahmoud Khamis[1], Abdallah Elashry[1], and Ayman Nada[1,**]

Abstract— **Robotic gimbals remain challenging to control due to sensor latency, nonlinear dynamics, and actuator limitations, which often lead to loss of lock when tracking fast-moving targets. In this work, a fast-moving target is one whose apparent displacement between consecutive video frames exceeds its image-space size, resulting in negligible inter-frame overlap and motion blur.**

This paper presents the design and experimental validation of a real-time 3-degree-of-freedom (3-DOF) gimbal system for robust high-speed target tracking using a vision and inertial measurement unit (IMU) velocity control framework. A YOLO-based object detection model localizes the target in the image plane, from which normalized image-plane errors relative to the camera center are computed. Roll angle feedback from an end-effector IMU maintains camera alignment.

Instead of position-based control, image-plane and roll errors are directly mapped to desired angular velocities in the camera frame. These velocities are regulated using three independent fuzzy proportional–integral–derivative (PID)-like controllers with Gaussian membership functions, one per axis. The angular velocity commands are transformed into joint velocities via the inverse Jacobian and executed using smart servo motors in internal velocity control mode over a controller area network (CAN) bus. Experimental results demonstrate stable lock-on, improved tracking accuracy, and robustness against latency and dynamic disturbances.

I. Introduction

Robotic gimbal systems stabilize sensors and enable visual target tracking on mobile platforms such as unmanned aerial vehicles (UAVs) [1], unmanned ground vehicles, and remotely operated vehicles. The gimbal must maintain continuous lock on a target [2] while the base platform experiences disturbances, vibrations, and unpredictable motion, a challenge that intensifies when the target exhibits fast, aggressive [3], and non-smooth motion.

Most gimbal tracking systems rely on position-based control, typically using classical proportional–integral–derivative (PID) controllers [4], [5]. While satisfactory for slow or moderate targets under stable conditions, their effectiveness degrades with high target velocities [6], sensing latency, and nonlinear multibody dynamics [8]. Position-based control accumulates tracking error, resulting in large corrective commands and oscillatory behavior when both target and platform move rapidly. In practical vision pipelines operating at 30–60 Hz, fast-moving targets may shift tens of pixels between consecutive frames, causing conventional position-based controllers to accumulate tracking error and generate delayed corrective responses.

Existing visual servoing approaches commonly rely on position regulation, predictive filtering, or model-based compensation to address target motion and sensing delay. However, these methods either require accurate dynamic models, introduce additional computational overhead, or exhibit reduced robustness under rapidly changing disturbances and aggressive target maneuvers. In embedded robotic platforms operating under strict real-time constraints, maintaining stable lock-on at high target velocities remains an open challenge.

To address these limitations, this paper proposes a velocity-based visual servoing framework for high-speed target tracking using a 3-DOF robotic gimbal. Unlike conventional image-based visual servoing approaches that primarily regulate position error or rely on predictive compensation, the proposed method directly maps image-plane errors to angular velocity commands in the camera frame, reducing delay-induced accumulation effects during aggressive target motion and platform disturbances. Image-plane tracking errors are converted into desired angular velocities, enabling faster corrective actions. Roll stabilization is achieved using end-effector IMU feedback, and a fuzzy PID-like control strategy with Gaussian membership functions handles nonlinearities and rapidly changing error dynamics, while preserving the computational efficiency required for embedded real-time mechatronic systems [7].

The proposed approach is validated on a physical testbed subjected to coupled base disturbances from a Stewart platform, demonstrating stable lock-on and low tracking error for fast targets with minimal oscillations and no delay-induced overshoot.

The main contributions are: (i) a velocity-space visual servoing formulation mapping image-plane errors directly to camera angular velocities, avoiding position-based accumulation errors; (ii) a fuzzy PID-like controller with Gaussian membership functions providing adaptive gain scheduling without an explicit dynamic model; and (iii) experimental validation on a physical 3-DOF gimbal under coupled base disturbances and high-speed target motion.

II. System Architecture

The proposed system consists of a 3-DOF robotic gimbal [9] for real-time visual target tracking under unstable platform conditions. The gimbal adopts a serial kinematic

[1]Mechatronics & Robotics Eng. Dept., School of Innovative Design Engineering, Egypt-Japan University of Science and Technology (E-JUST), Alexandria, Egypt.
[*]Ahmed Attia (ahmed.attia@ejust.edu.eg)
[**]Ayman Ali Nada (ayman.nada@ejust.edu.eg)

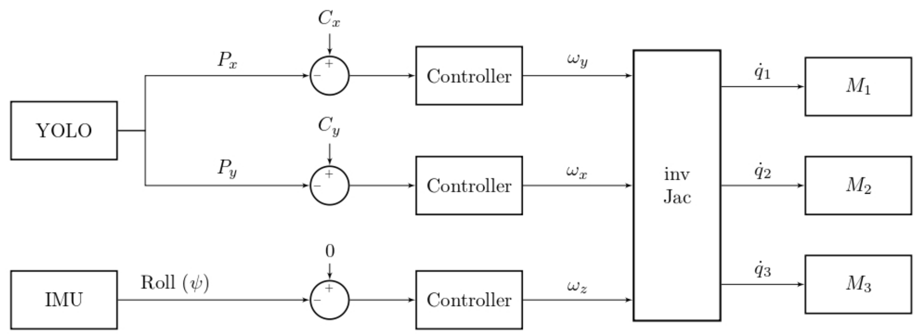

Fig. 1. Block diagram of the proposed 3-DOF gimbal system architecture, showing the flow from perception to control and actuation.

configuration with three rotational joints providing yaw, pitch, and roll motion. A monocular RGB camera and an IMU are rigidly mounted at the end effector, ensuring visual and inertial sensing share the same reference frame.

Actuation uses MG4010E-i10v3 smart servo motors with dual encoders and internal velocity control loops. The motors operate in velocity control mode, allowing direct joint angular velocity commands. All motors communicate via a CAN bus for deterministic, low-latency command transmission.

The processing pipeline runs on a Raspberry Pi 3, which captures video from the Pi Camera Module V2 and streams it via UDP over a local network to an external laptop running the YOLOv8 inference engine. The video stream is handled through a GStreamer pipeline to minimize latency, and the detection model is exported to TensorRT format for accelerated GPU inference. Detected bounding box coordinates are sent back to the Raspberry Pi, which reads the BNO055 IMU over I^2C, computes the control law, and transmits joint velocity commands to the servo motors via the CAN bus. The high-level control loop integrates visual perception, inertial sensing, and control computation in real time. Image-plane errors and IMU roll feedback are converted to desired angular velocities in the camera frame, then mapped to joint-space commands through the inverse Jacobian (Section III-C). The system architecture is shown in Fig. 1.

III. Gimbal Modeling and Control Architecture

A. Preliminaries: Fuzzy Logic and Gaussian Membership Functions

The controller uses fuzzy logic inference to map numerical inputs to control outputs through linguistic rules. Each variable is associated with fuzzy sets characterized by membership functions quantifying the degree of belonging to each linguistic category.

Gaussian membership functions are employed for all fuzzy sets:

$$\mu(x) = \exp\left(-\frac{(x-c)^2}{2\sigma^2}\right), \tag{1}$$

where c is the center and σ is the width. Gaussian functions are preferred over triangular or trapezoidal alternatives because they are smooth and differentiable, ensuring continuous control output without abrupt transitions that could excite mechanical resonances.

Five linguistic terms are defined for each variable: NB (Negative Big), NS (Negative Small), ZE (Zero), PS (Positive Small), and PB (Positive Big). These span the universe of discourse at evenly spaced points, with NB and PB representing large-magnitude errors, NS and PS moderate deviations, and ZE values near zero. This partition provides sufficient granularity while keeping the rule base compact.

The fuzzy inference system uses a Mamdani-type architecture with min–max composition and centroid defuzzification. For each control cycle, the crisp input values $(e, \Delta e)$ are fuzzified by evaluating their membership degrees across all five fuzzy sets. The rule base is then evaluated using the min operator to determine the firing strength of each rule, and the max operator aggregates the consequent fuzzy sets. The final crisp output ω_f is obtained through centroid defuzzification, which computes the center of gravity of the aggregated output membership function. The rule base follows a monotonic structure where larger errors and faster error rates produce stronger control actions with smooth transitions near equilibrium, as shown in Table II.

B. Visual Target Detection and Error Representation

The visual perception subsystem detects targets and computes controller error signals. A fine-tuned YOLOv8 model [10] detects tennis ball targets in real time. The camera is a Raspberry Pi Camera Module V2, capturing 640×480 video at 60 Hz. The video stream is transmitted via UDP and processed on an external laptop using GStreamer, with the model exported to TensorRT format to reduce inference latency. The TensorRT optimization reduces the per-frame inference time to approximately 8 ms, enabling the overall control loop to maintain a 30 ms cycle time despite the network round-trip delay.

Once detected, the bounding box center $(x_{\text{obj}}, y_{\text{obj}})$ is extracted. The image-plane error relative to the image center $(x_c, y_c) = (320, 240)$ is

$$e_x = x_{\text{obj}} - x_c, \quad e_y = y_{\text{obj}} - y_c. \tag{2}$$

Errors are normalized to $[-1,1]$ and differentiated to obtain the error rate, forming the controller input [11].

The roll angle is measured using a BNO055 IMU at 100 Hz and normalized from $[-90°, 90°]$ to $[-1,1]$. The three-dimensional error vector

$$\mathbf{e} = [e_x, e_y, e_{\text{roll}}]^T, \tag{3}$$

serves as input to the velocity-based controller (Section III-D). Each component is processed by an independent fuzzy PID-like controller instance, producing angular velocity commands ω_x, ω_y, and ω_z in the camera frame. The decoupled control structure simplifies tuning and reduces computational overhead compared to a fully coupled multi-input multi-output fuzzy system.

C. Gimbal Kinematics and Velocity Mapping

The 3-DOF gimbal has three rotational joints in a serial yaw–pitch–roll configuration. The coordinate frames are shown in Fig. 2.

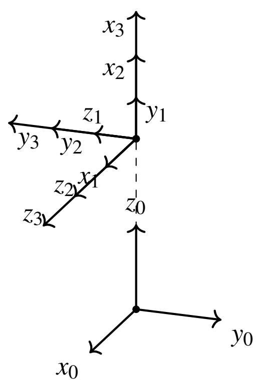

Fig. 2. Frame configuration of the 3-DOF gimbal, showing joint axes and end-effector orientation.

The Denavit–Hartenberg parameters are listed in Table I.

TABLE I
DENAVIT–HARTENBERG PARAMETERS OF THE 3-DOF GIMBAL

Link	θ_i	d_i (mm)	a_i (mm)	α_i
1	θ_1	237	0	90°
2	θ_2	0	0	90°
3	θ_3	0	0	0°

The rotation matrix of the end-effector frame relative to the base, aligned with the camera coordinate system, is

$$R_{0\to\text{camera}} = \begin{bmatrix} c_1c_2s_3 - c_3s_1 & s_1s_3 + c_1c_2c_3 & c_1s_2 \\ c_1c_3 + c_2s_1s_3 & c_2c_3s_1 - c_1s_3 & s_1s_2 \\ s_2s_3 & c_3s_2 & -c_2 \end{bmatrix}, \tag{4}$$

where $c_i = \cos(\theta_i)$ and $s_i = \sin(\theta_i)$. The angular velocity Jacobian is

$$J_\omega = \begin{bmatrix} 0 & s_1 & c_1s_2 \\ 0 & -c_1 & s_1s_2 \\ 1 & 0 & -c_2 \end{bmatrix}, \tag{5}$$

with inverse

$$J_\omega^{-1} = \begin{bmatrix} \frac{c_1c_2}{s_2} & \frac{s_1c_2}{s_2} & 1 \\ s_1 & -c_1 & 0 \\ \frac{c_1}{s_2} & \frac{s_1}{s_2} & 0 \end{bmatrix}. \tag{6}$$

The joint velocities are computed from the desired camera angular velocity $\omega = [\omega_x, \omega_y, \omega_z]^T$ as

$$\dot{q} = J_\omega^{-1} R_{0\to\text{camera}}\, \omega = \begin{bmatrix} \frac{\omega_x s_3 + \omega_y c_3}{s_2} \\ \omega_y s_3 - \omega_x c_3 \\ \frac{\omega_x s_3 c_2 + \omega_y c_3 c_2}{s_2} + \omega_z \end{bmatrix}. \tag{7}$$

This maps camera angular velocities to joint-space motor commands for real-time actuation.

D. Control Architecture

The control architecture combines fuzzy control with velocity-level kinematic mapping, outputting angular velocity commands directly to the gimbal actuators.

1) PD-like Fuzzy Control: The PD-like fuzzy controller takes the normalized tracking error $e \in [-1,1]$ and its derivative $\Delta e \in [-15,15]$ s^{-1} as inputs. The error derivative is computed using a first-order backward difference.

The fuzzy inference system maps $(e, \Delta e)$ to an intermediate angular velocity command ω_f using the rule base in Table II. This provides proportional action through the error and derivative action through the error rate, enabling fast response and oscillation damping without an explicit model. The controller structure is shown in Fig. 3.

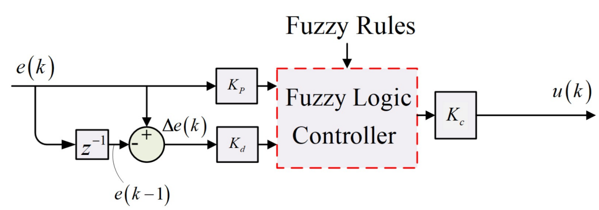

Fig. 3. PD-like fuzzy controller with error and error-derivative inputs.

2) PI-like Integral Augmentation: While the PD-like controller ensures rapid transient response, steady-state errors may persist under constant disturbances. An integral action is introduced outside the fuzzy inference mechanism [12]:

$$\omega_I(t) = \omega_I(t-1) + k_i \cdot \omega_f(t), \tag{8}$$

where k_i is the integral gain. The accumulated fuzzy output compensates for persistent tracking errors, as illustrated in Fig. 4.

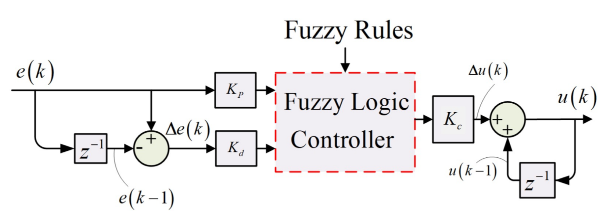

Fig. 4. PI-like control obtained by integrating the fuzzy PD output.

3) PID-like Fuzzy Control Structure: The final control signal combines the proportional and integral terms:

$$\omega_P(t) = k_p \cdot \omega_f(t), \tag{9}$$

$$\omega(t) = \omega_P(t) + \omega_I(t), \tag{10}$$

with gains $k_p = 1.0$ and $k_i = 0.1$. This PID-like structure preserves nonlinear adaptability while achieving zero steady-state error. The complete controller is shown in Fig. 5.

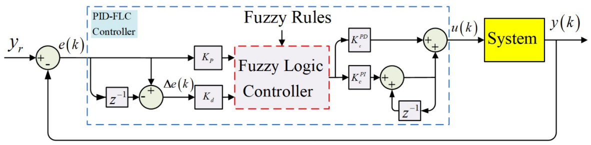

Fig. 5. PID-like fuzzy controller combining PD and PI actions.

4) Membership Functions and Rule Base: The universes of discourse are $e \in [-1, 1]$, $\Delta e \in [-15, 15]$ s^{-1}, and $\omega_f \in [-150, 150]$ deg/s. The Gaussian membership functions are shown in Figs. 6–8. The rule base (Table II) follows a monotonic mapping where larger errors and faster rates produce stronger corrective actions.

TABLE II
FUZZY RULE BASE FOR PID-LIKE CONTROLLER

$\Delta e \backslash e$	NB	NS	ZE	PS	PB
NB	NB	NB	NS	ZE	PS
NS	NB	NS	NS	ZE	PS
ZE	NS	NS	ZE	PS	PS
PS	NS	ZE	PS	PS	PB
PB	ZE	PS	PS	PB	PB

IV. Experimental Validation

Experiments were conducted on a testbed to evaluate the proposed velocity-space fuzzy control framework under dynamic conditions representative of mobile robotic platforms.

A. Experimental Setup and Disturbance Generation

The 3-DOF gimbal was mounted on a Stewart platform (6-DOF parallel manipulator) capable of generating coupled translational and rotational disturbances, emulating the vibrations and attitude variations encountered in unmanned aerial and ground vehicles. The hardware setup is shown in Fig. 9. The Stewart platform operated using its dedicated control and data acquisition system to generate predefined disturbance trajectories in real time, while the gimbal control loop ran independently on the Raspberry Pi 3. The camera and IMU were mounted directly on the gimbal end-effector, ensuring that visual and inertial measurements reflected the stabilized sensor frame during operation.

To stress the controller under realistic multi-axis disturbances, the Stewart platform generated synchronized rotational and translational oscillations. The rotational disturbances emulate attitude perturbations commonly experienced

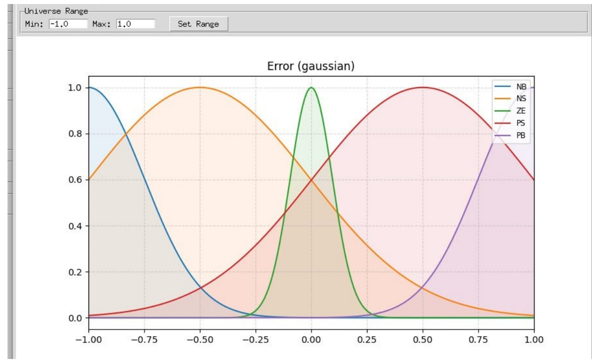

Fig. 6. Gaussian membership functions for the normalized error e.

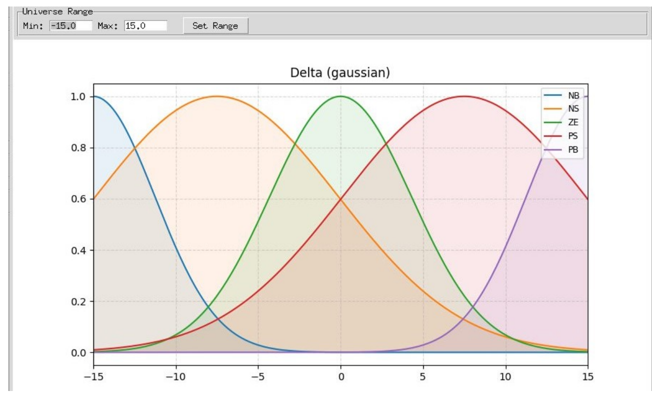

Fig. 7. Gaussian membership functions for the error derivative Δe.

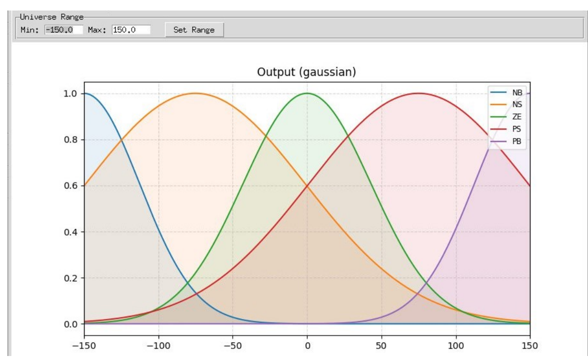

Fig. 8. Gaussian membership functions for the controller output ω_f.

by mobile robotic platforms, while the translational disturbances reproduce base vibrations and positional fluctuations. The rotational motion profile was defined as:

$$\left.\begin{aligned} \theta_z(t) &= A_\theta \sin(\phi)\sin(t), \\ \theta_x(t) &= A_\theta \cos(\phi)\sin(t), \\ \theta_y(t) &= 0 \end{aligned}\right\} \tag{11}$$

where $A_\theta = -0.042$ rad and $\phi = 2.36$ rad is the phase angle. These equations generate coupled sinusoidal roll and pitch motions with a fixed phase relationship. The translational

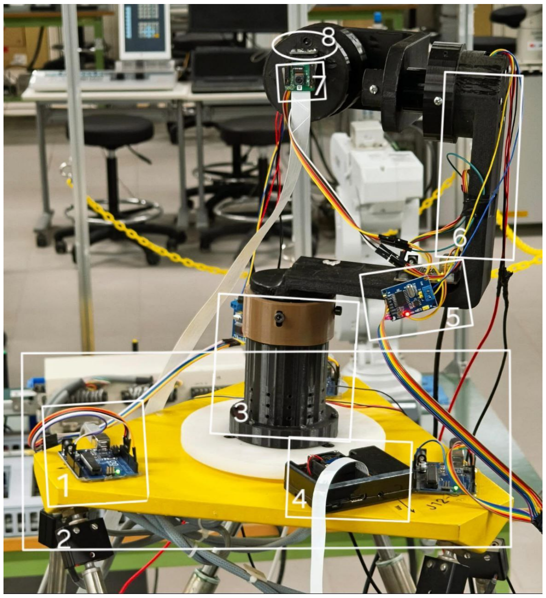

Fig. 9. Experimental test-rig: (1) Arduino UNO, (2) Stewart Platform, (3) Gimbal base link, (4) Raspberry Pi 3, (5) CAN module, (6) Gimbal 2nd link, (7) Pi Camera V2, (8) BNO055.

motion was generated as:

$$\left.\begin{aligned} x_m(t) &= K_x \cos(\phi)\sin(t), \\ y_m(t) &= -K_x \sin(\phi)\sin(t), \\ z_m(t) &= 100 + 37.5\cos(t) \end{aligned}\right\} \tag{12}$$

with $K_x = 75$ mm. The simultaneous excitation of translational and rotational axes produces nonlinear disturbance coupling at the gimbal base, creating a challenging tracking environment with continuously varying target motion relative to the camera frame. The resulting base orientation profiles are shown in Fig. 10.

The generated roll and pitch oscillations introduce persistent perturbations that continuously alter the camera orientation, thereby evaluating the robustness of the proposed velocity-based controller against sensing latency, inertial coupling, and rapid directional changes.

B. Performance Metrics and Scenarios

The control loop operated at 30 ms (33 Hz). Performance was evaluated using the tracking error (Euclidean distance between target center and image frame center) and the normalized mean squared error (MSE) of image-plane coordinates. Two scenarios were tested.

In the first, a static target was tracked while the Stewart platform perturbed the base, validating stabilization and IMU-based roll correction. This scenario isolates the gimbal's disturbance rejection capability by removing target motion as a variable.

In the second, the target was suspended as a pendulum released from 30°, inducing fast harmonic motion over 22 seconds (10 oscillation cycles) with simultaneous base disturbances. This double-dynamic scenario tests tracking with both pursuer and target in motion, representing the most

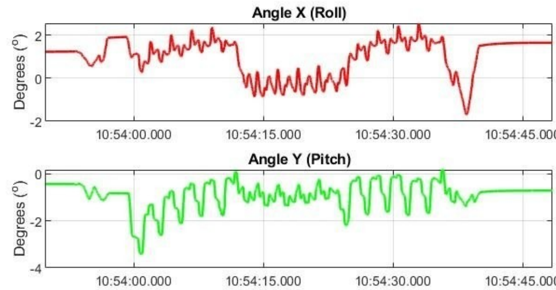

Fig. 10. Measured roll and pitch disturbance profiles generated by the Stewart platform acting on the gimbal base.

demanding operating condition. The pendulum frequency was approximately 0.45 Hz, producing peak angular velocities that exceeded the gimbal's comfortable tracking range, deliberately stressing the velocity control framework.

C. Results and Analysis

The stationary target results (Fig. 11) show the controller maintaining the target near the image center despite significant base disturbances. After initial lock-on, the system demonstrated a settling time under 200 ms, and the steady-state error remained bounded within a narrow margin, confirming effective disturbance rejection. The horizontal axis exhibits slightly larger oscillations than the vertical axis, consistent with the yaw joint bearing the greater inertial load of the gimbal assembly.

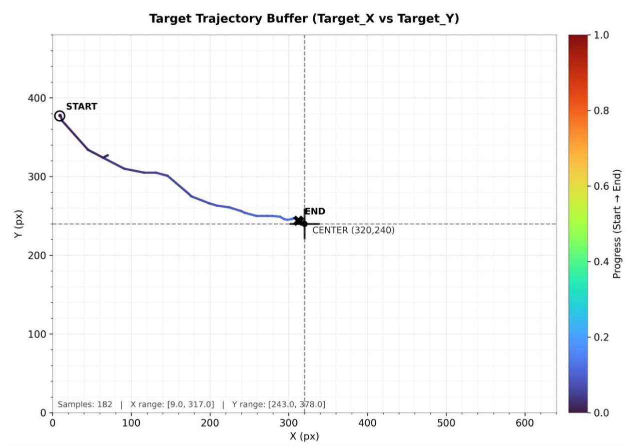

Fig. 11. Scenario A results: time-domain response of the target centroid coordinates while tracking a stationary object under active base disturbances.

The dynamic tracking results (Fig. 12) show the target coordinates clustered around the image center throughout the 22-second trial. The gimbal effectively matches the pendulum velocity despite aggressive direction reversals at each oscillation extreme. The normalized MSE was 2.98×10^{-3} (horizontal) and 1.84×10^{-4} (vertical), confirming robust lock-on with minimal phase lag under combined dynamics. The higher horizontal MSE reflects the greater yaw-axis disturbance coupling from the Stewart platform, which primarily excited roll and pitch base motions that project more strongly onto the gimbal's horizontal image axis.

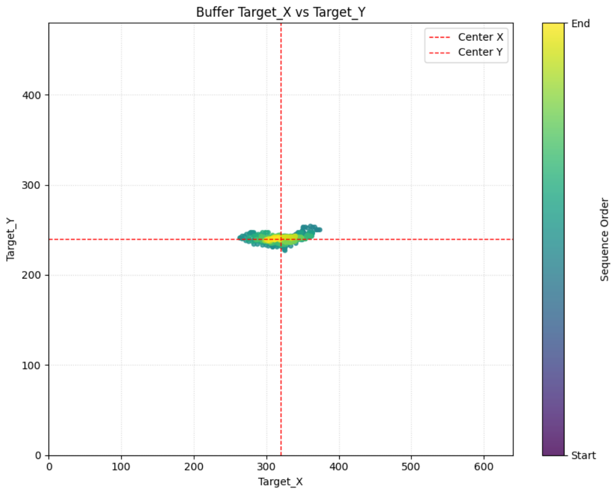

Fig. 12. Scenario B results: tracking performance during the double-dynamic test over 10 full oscillations. The normalized MSE is 2.98×10^{-3} (X-axis) and 1.84×10^{-4} (Y-axis). Axes are in pixels (px) for a 640×480 image frame.

V. Discussion

Velocity-space control effectively addresses perception latency in vision-based tracking. The 30 ms loop, including image acquisition, YOLO inference, and control computation, would introduce significant phase lag in a position-based scheme. Commanding angular velocities compensates for this delay through rate-based correction, evident in the tight tracking data clustering during the dynamic scenario.

The fuzzy controller provides two operating regimes unavailable to fixed-gain PID. During large transients, outer membership functions (NB or PB) produce strong corrective commands, yielding sub-200 ms settling. Near equilibrium, Gaussian overlap between adjacent membership functions ensures gradual gain reduction, preventing oscillation from sensor noise. This explains the improved disturbance rejection in the stationary target scenario.

Two limitations are noted. First, gains were selected empirically, limiting reproducibility across configurations. Second, the inverse Jacobian has a singularity at zero pitch angle, avoided here by mechanical workspace constraints.

VI. Conclusion

This paper presented a velocity-based visual servoing framework for a 3-DOF robotic gimbal addressing sensing latency and nonlinear dynamics in high-speed target tracking. Mapping image-plane errors directly to angular velocity commands eliminates the error accumulation degrading position-based approaches under fast motion and base disturbances. The fuzzy PID-like controller with Gaussian membership functions provides nonlinear gain adaptation without an explicit dynamic model, and the inverse Jacobian enables real-time camera-frame to joint-space velocity transformation.

Experimental validation with a Stewart platform generating coupled base disturbances confirmed the approach, achieving sub-200 ms settling for stationary targets and robust lock during combined target and platform motion, with normalized MSE of 2.98×10^{-3} and 1.84×10^{-4} for the horizontal and vertical axes.

The results demonstrate that velocity-space fuzzy control provides an effective and computationally efficient alternative to model-based and learning-based tracking approaches for embedded robotic vision systems operating under severe dynamic disturbances and sensing latency. The proposed framework shows strong potential for deployment in autonomous aerial and mobile robotic platforms requiring robust real-time visual target tracking.

Future work will integrate predictive filtering through an Extended Kalman Filter for trajectory estimation during occlusions, field deployment on a UAV for real-world aerodynamic disturbance evaluation, systematic gain optimization replacing empirical tuning, and exploration of deep reinforcement learning for end-to-end policy learning in highly nonlinear tracking scenarios.

References

[1] Ö. Villi and H. Yavuz, "The utilization of gimbal systems in unmanned aerial vehicles," *Advanced UAV*, vol. 4, pp. 19–30, June 2024.

[2] R. Cunha, M. Malaca, V. Sampaio, B. J. N. Guerreiro, P. Nousi, I. Mademlis, A. Tefas, and I. Pitas, "Gimbal Control for Vision-based Target Tracking," in *Proc. EUSIPCO Satell. Workshop Signal Process., Comput. Vis. Deep Learn. Auto. Syst.*, 2019.

[3] D. Rozumnyi, J. Kotera, F. Sroubek, L. Novotny, and J. Matas, "The World of Fast Moving Objects," in *Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, July 2017, pp. 4838–4846.

[4] M. Şahin, "Stabilization of Two Axis Gimbal System with Self Tuning PID Control," *J. Polytechnic*, vol. 27, no. 4, pp. 1441–1452, 2024.

[5] Y. Zhou, Y. Li, B. Wang, and Y. Ma, "Real-Time Gimbal Control for Target Tracking Using Object Detection and PID Tuning," in *Proc. 2020 IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, 2020, pp. 433–437.

[6] Q. Wu, Z. Yu, C. Xu, and T. Zhou, "Drone-Net: A High-Precision Real-Time Gimbal Tracking System for Low-Slow-Small Drone Detection Based on Enhanced Faster R-CNN," in *Proc. 2023 Int. Conf. Image Process., Comput. Vis. Mach. Learn. (ICICML)*, Chengdu, China, 2023.

[7] A. A. Nada and M. A. Bayoumi, "Development of embedded fuzzy control using reconfigurable FPGA technology," *Automatika*, vol. 65, no. 2, pp. 609–626, 2024.

[8] A. A. Nada and A. H. Bishiri, "Multibody system design based on reference dynamic characteristics: gyroscopic system paradigm," *Mechanics Based Design of Structures and Machines*, vol. 51, no. 6, pp. 3372–3394, 2023.

[9] X. Liu, Y. Yang, C. Ma, J. Li, and S. Zhang, "Real-Time Visual Tracking of Moving Targets Using a Low-Cost Unmanned Aerial Vehicle with a 3-Axis Stabilized Gimbal System," *Appl. Sci.*, vol. 10, no. 15, p. 5064, 2020.

[10] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO," version 8.0.0, Ultralytics, Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[11] D. J. Regner, J. D. Salazar, P. V. Buschinelli, M. Machado, D. Oliveira, J. M. Santos, C. A. Marinho, and T. C. Pinto, "Object Tracking Control Using a Gimbal Mechanism," *ISPRS Archives*, vol. XLIII-B1, pp. 189–196, 2021.

[12] A. Fathi, A. M. Kamel, A. O. Elfarouk, and Y. Z. Elhalwagy, "Modelling and simulation of two axes gimbal fuzzy PI stabilization system in the presence of feedback sensors noise," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1172, p. 012035, 2021.