

Co-Simulation of the Agile Eye Robot Using MSC.ADAMS and MATLAB: Design, Kinematics, and Control

Mohammad Heidar Khamsehei Fadaei

moh13359@gmail.com

Islamic Azad University Saveh <https://orcid.org/0000-0001-6425-0728>

Fahimeh Sadat Ghasemi

Qazvin Islamic Azad University <https://orcid.org/0009-0001-2397-0937>

Research Article

Keywords: Parallel manipulator, Agile Eye, co-simulation, MSC.ADAMS, MATLAB, inverse kinematics, Fuzzy Adaptive PID controller

Posted Date: September 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11091510/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Co-Simulation of the Agile Eye Robot Using MSC.ADAMS and MATLAB: Design, Kinematics, and Control

Mohammad Heidar Khamsehei Fadaei ¹, and Fahimeh Sadat Ghasemi ²

Abstract: This paper proposes a fuzzy adaptive PID controller for the Agile Eye spherical parallel robot and evaluates its performance within a co-simulation framework integrating MSC.ADAMS and MATLAB/Simulink. The main objective is to quantify the advantages of online gain scheduling over fixed PID gains for this overconstrained mechanism. A detailed mechanical model incorporating constraints, actuators, and sensors is built directly in ADAMS and exported to Simulink as a nonlinear plant, thereby preserving the actual mass, inertia, and frictional characteristics. Closed-form inverse kinematics are derived and validated against forward kinematics through a round-trip test, yielding an orientation reconstruction error on the order of 10^{-4} degrees. Three controllers are implemented: a PD controller, a fixed-gain PID tuned via the Ziegler–Nichols method, and a fuzzy adaptive PID that employs 49 rules to adjust its gains online. Their performance is compared under step commands, a smooth planned path, and square-wave signals. The results demonstrate that the fuzzy adaptive controller consistently outperforms its fixed-gain counterparts reducing the IAE by roughly 60%, keeping overshoot below 5%, and virtually eliminating steady-state error. Overall, the co-simulation workflow proves to be a practical and effective means of evaluating control strategies without requiring a physical prototype.

Keywords: Parallel manipulator, Agile Eye, co-simulation, MSC.ADAMS, MATLAB, inverse kinematics, Fuzzy Adaptive PID controller.

I. INTRODUCTION

Parallel kinematic machines (PKMs) have long attracted considerable interest in applications requiring high structural stiffness, low moving inertia, and accurate multi axis motion. In particular, spherical parallel manipulators (SPMs) are well suited for pure rotational tasks around a fixed point, such as camera orientation systems, motion simulators, and rehabilitation devices [1]–[3]. Other applications include the use of parallel mechanisms in robotic machining beds, where accurate position control is critical [4]. Although SPMs have elegant and compact architectures, they still suffer from persistent practical implementation and control difficulties, such as strong dynamic coupling between actuated axes, configuration dependent inertia variations, and unmodelled friction or link flexibility.

The Agile Eye, originally proposed by Gosselin and Angeles in the early 1990s and further developed in subsequent studies [5], [6], has become a benchmark mechanism in the SPM literature. The mechanism has three degrees of freedom and consists of three identical RRR kinematic chains intersecting at

a common centre of rotation. Its inverse kinematics has a closed form solution, which greatly facilitates trajectory planning [7]. Conversely, its forward kinematics requires the solution of a system of nonlinear equations, often needing a careful initial guess [7]. This asymmetry between the inverse and forward problems is typical of parallel robots, but it also implies that any simulation-based control design must consider both aspects rigorously.

In the early stages of this work, we recognised that kinematics alone is not sufficient for high-performance control. The Agile Eye has highly nonlinear and configuration dependent dynamics, which render purely kinematic controllers inadequate. The standard method of deriving the full equations of motion from Lagrangian mechanics, followed by linearisation, is time consuming and sensitive to parametric changes. This motivated us to adopt a co-simulation framework, in line with recent practice [8]–[10]. Specifically, we constructed the entire mechanical assembly directly in MSC.ADAMS, specified realistic material properties and joint constraints, and exported it to MATLAB/Simulink as a nonlinear plant. This approach preserves the exact inertia, geometry, and friction characteristics while allowing visual feedback through the ADAMS viewer during the control loop providing a realistic testing environment without the expense of physical hardware.

On the control side, fixed gain PID regulators remain the most common in industrial practice. However, for a highly nonlinear manipulator such as the Agile Eye, a single fixed gain set rarely provides satisfactory performance throughout the workspace. It often forces a compromise between excessive overshoot during rapid manoeuvres and sluggish responses near workspace singularities. Fuzzy logic offers a systematic and practical approach for online gain scheduling, where the PID parameters are adjusted in real time according to the instantaneous error and its derivative. A comprehensive review of different control approaches for parallel robots is presented in [11], ranking methods from classical PID to advanced adaptive strategies. The effectiveness of fuzzy adaptive PID controllers has recently been shown for serial manipulators and mobile robots [12]–[16]. Robust adaptive control approaches have also been experimentally validated on spherical parallel robots, especially under insufficient excitation conditions [17]. Nevertheless, to the best of our knowledge, no previous study has rigorously applied such a controller to the Agile Eye and

¹ Corresponding author. Department of Mechanical Engineering, Islamic Azad University, Saveh, Iran, Email: moh13359@gmail.com

² Department of Computer Engineering, Islamic Azad University, Qazvin, Iran.

performed a comprehensive quantitative comparison against classical PD and PID controllers under multiple excitation signals.

The main objective of this paper is to answer the following question: what quantitative improvement does fuzzy gain scheduling offer for a spherical parallel robot when evaluated in a high-fidelity co-simulation environment? To address this, we (i) developed a detailed ADAMS model, verified its static torque against an analytical estimate, and exported it to Simulink; (ii) derived closed form inverse kinematics and validated them through round trip forward kinematic tests, achieving orientation errors on the order of 10^{-4} degrees; (iii) designed and implemented three control architectures PD, fixed gain PID (tuned via Ziegler–Nichols), and a 49 rule fuzzy adaptive PID within the same loop; and (iv) compared their performances under smooth planned trajectories, square wave references, and step inputs, using IAE, ISE, overshoot, settling time, and steady state error as quantitative metrics.

For the sake of full reproducibility, we also discuss our practical design choices, such as the reference model time constant and the normalisation ranges of fuzzy inputs. We do not offer a formal stability proof for the adaptive scheme but instead rely on the boundedness of the scheduled gains and the inherent dissipativity of the mechanical plant, noting that a rigorous stability analysis remains a direction for future research.

The remainder of this paper is organised as follows. Section 2 describes the mechanical architecture and the construction of the ADAMS model. Section 3 presents the kinematic analysis and validation. Section 4 details the controller design and implementation. Section 5 reports the simulation results and discussion. Finally, Section 6 concludes the paper and points out directions for future work.

II. MECHANICAL ARCHITECTURE AND DESIGN

The Agile Eye is a spherical parallel manipulator with three degrees of freedom. Its mechanical structure consists of a fixed base, three identical lower links, three identical upper links, and a moving end effector (Fig. 1). Several variants of this mechanism exist, including a compliant version designed to minimise parasitic displacements [18]. In each kinematic chain, a lower link is attached to the base, and an upper link is attached to the end effector; all joints are revolute. The axes of all revolute joints intersect at a common point, denoted as the centre of rotation O . This geometric property ensures that the end effector undergoes pure rotation about O without any translation. The only actuated joints are those connecting the lower links to the base; the remaining joints are passive and serve only to constrain the end effector's orientation.

Please use automatic hyphenation and check your spelling. Additionally, be sure your sentences are complete and that there is continuity within your paragraphs. Check the numbering of your graphics (figures and tables) and make sure that all appropriate references are included.

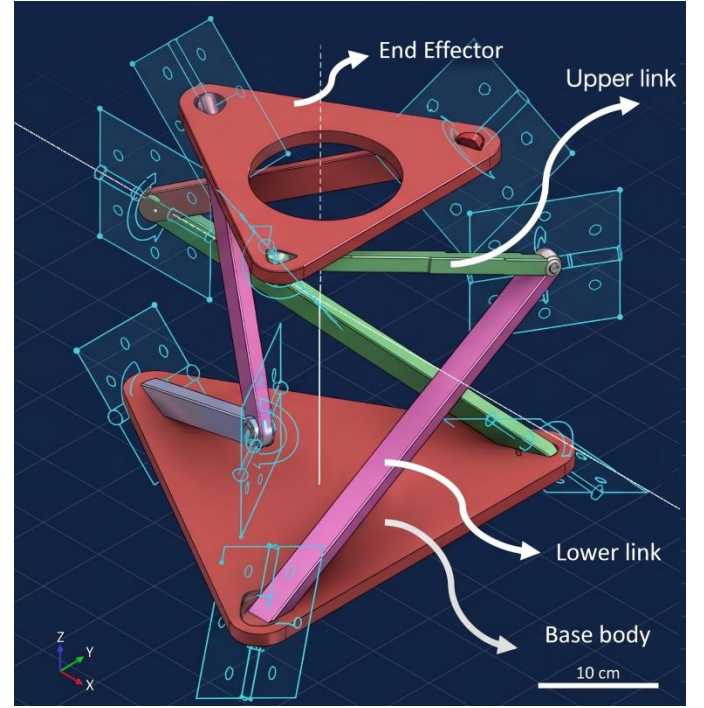


Fig. 1. Kinematic model of the Agile Eye manipulator.

The mechanical assembly is modelled directly in MSC.ADAMS, resulting in a high-fidelity dynamic model. In this environment, parts are assigned material properties, revolute and fixed constraints are applied, and actuation variables are defined. This modelling approach preserves the exact three-dimensional geometry and inertial characteristics of the physical prototype, yielding a virtual model that behaves realistically under simulated loads and motions. The ADAMS model serves both as a kinematic verification tool and as the plant to be exported to MATLAB/Simulink for controller implementation.

Table 1. Physical parameters of the Agile Eye manipulator

Parameter	Symbol	Value	Unit
Base radius	R_b	0.120	m
Lower link length	L_l	0.180	m
Upper link length	L_u	0.210	m
End-effector radius	R_e	0.080	m
Mass of lower link	m_l	0.52	kg
Mass of upper link	m_u	0.38	kg
Mass of end-effector	m_e	0.65	kg
Moment of inertia of lower link	I_l	1.4×10^{-3}	$\text{kg} \cdot \text{m}^2$
Moment of inertia of upper link	I_u	9.8×10^{-4}	$\text{kg} \cdot \text{m}^2$
Moment of inertia of end-effector	I_e	2.1×10^{-3}	$\text{kg} \cdot \text{m}^2$
Material	—	Aluminium 6061-T6	—
Coefficient of viscous friction at joints	b	0.015	$\text{N} \cdot \text{m} \cdot \text{s} / \text{rad}$

A right-handed orthogonal coordinate system $\{O; X, Y, Z\}$ is attached to the stationary base, with its origin at the rotation centre O . The Z -axis is perpendicular to the base plane and points upward. The Y -axis lies in the plane formed by the Z -axis and the u_1 -axis of the first actuated joint (Fig. 2). The three

actuated joints are symmetrically arranged around the base at 120° intervals. The orientation of the i -th actuated joint axis ($i = 1, 2, 3$) is represented by the unit vector u_i . The intermediate passive joint linking the lower arm to the upper arm is denoted as w_i , and the unit vector along the distal joint connected to the end-effector is denoted as v_i .

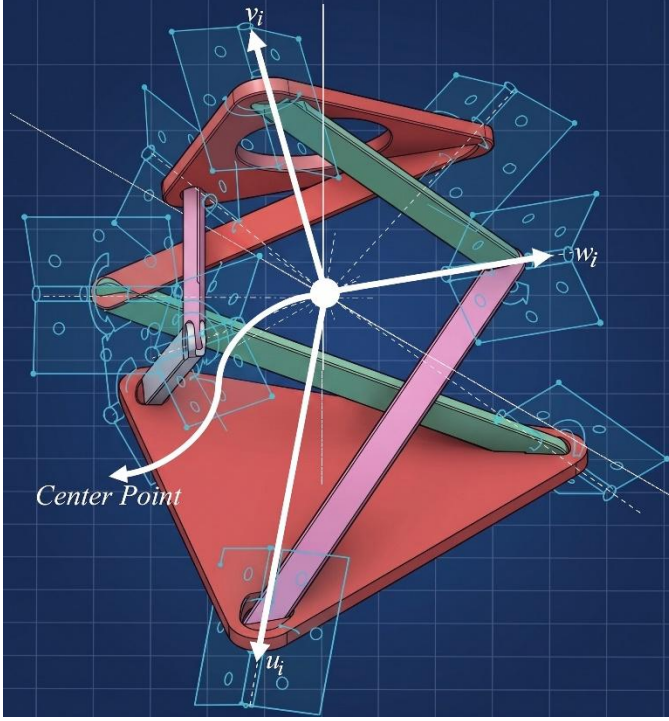


Fig. 2. Direction vectors of the joints in one kinematic chain

The geometry of each limb is defined by two fixed angles. Angle γ is the inclination of u_i with respect to the Z-axis, and angle β is the inclination of v_i . In our manipulator, both angles are set to 54.75° to achieve a symmetric and isotropic workspace. Further optimal design studies have refined these geometric parameters to enhance kinetostatic performance [19].

The azimuth angles η_i define the plane containing the i -th actuated joint axis relative to the Y-axis:

$$\eta_i = \frac{2(i-1)\pi}{3} \quad (i = 1, 2, 3) \quad (1)$$

Moreover, the angles between successive joint axes within one limb are fixed: α_1 is the angle between u_i and w_i , and α_2 is the angle between w_i and v_i . These constant twist angles are determined by the link geometry.

The direction vectors u_i and w_i can be expressed in the base coordinate frame as functions of the input joint angles θ_i (the rotation of the lower link about u_i) and the constant geometric parameters:

$$u_i = \begin{bmatrix} \sin \eta_i \sin \gamma \\ \cos \eta_i \sin \gamma \\ -\cos \gamma \end{bmatrix} \quad (2a)$$

$$w_i = \begin{bmatrix} \sin \eta_i \sin \gamma \cos \alpha_1 - (\cos \eta_i \sin \theta_i - \sin \eta_i \cos \gamma \cos \theta_i) \sin \alpha_1 \\ \cos \eta_i \sin \gamma \cos \alpha_1 + (\sin \eta_i \sin \theta_i + \cos \eta_i \cos \gamma \cos \theta_i) \sin \alpha_1 \\ -\cos \gamma \cos \alpha_1 + \sin \gamma \cos \theta_i \sin \alpha_1 \end{bmatrix} \quad (2b)$$

These expressions are used in the kinematic analysis to relate the input angles to the end-effector orientation.

A preliminary mobility assessment is performed using the classical Kutzbach-Grübler criterion:

$$F = \lambda(n - j - 1) + \sum_{i=1}^j f_i \quad (3)$$

Where F is the degree of freedom, λ is the motion space dimension ($\lambda=6$ for unrestricted spatial motion), n is the number of links, j is the number of joints, and f_i is the connectivity of the i -th joint. Substituting $n=8$ (base, three lower links, three upper links, and end-effector), $j=9$ (all revolute joints), and $\sum f_i=9$ gives:

$$F = 6(8 - 9 - 1) + 9 = -3 \quad (4)$$

The negative result indicates that the Kutzbach-Grübler formula is not directly applicable to overconstrained mechanisms with special geometries, as it does not account for redundant constraints. Using screw theory, the true mobility of the Agile Eye can be shown to be three. These three independent degrees of freedom correspond to the orientation angles of the end-effector, typically parameterised by azimuth, tilt, and torsion (ϕ , θ , and ψ), as described later in the kinematics section.

A. Preliminary Validation of the Dynamic Model

This To verify that the ADAMS model accurately captures the rigid-body dynamics of the Agile Eye, we perform a simple static check. Recently, dynamic response analysis of similar spherical parallel manipulators has been studied for kinematic compensation on uneven terrain [20]. The manipulator is placed in the home configuration (all input angles $\theta_i = 0^\circ$, meaning the end-effector points vertically upward), and the actuated joints are locked. In this configuration, the weight of the moving components generates a static torque about each actuated joint. This torque is estimated by a first-order static analysis and compared with the value computed by the ADAMS solver.

Owing to the symmetric structure, the static torque τ_{static} on each actuator in the home pose is mainly caused by the weight of the upper link and part of the end-effector mass. Approximating the centres of mass at the geometric midpoints of the upper link and the end-effector platform gives:

$$\tau_{\text{static}} \approx \left(m_u g \frac{L_u}{2} + \frac{m_e}{3} g R_e \right) \sin \gamma \quad (5)$$

where $g = 9.81 \text{ m/s}^2$ and $\gamma = 54.75^\circ$. Substituting the numerical values from Table 1 yields:

$$\begin{aligned} \tau_{\text{static}} &\approx \left(0.38 \times 9.81 \times 0.105 + \frac{0.65}{3} \times 9.81 \times 0.080 \right) \times \sin 54.75^\circ \\ &\approx (0.391 + 0.170) \times 0.816 \\ &\approx 0.458 \text{ N} \cdot \text{m}. \end{aligned} \quad (6)$$

The same configuration was simulated in MSC.ADAMS with gravity enabled and the joints locked. The static torque measured at each actuated joint is $0.472 \text{ N} \cdot \text{m}$. The difference between the analytical estimate and the ADAMS value is about 3%, which is well within the tolerance expected from the simplified centre-of-mass assumptions. This close agreement confirms that the mass properties, joint constraints, and gravity model were implemented correctly in the ADAMS model. This geometrically precise and kinematically validated mechanical model, built directly in ADAMS, forms the foundation for all subsequent kinematic derivations, dynamic simulations, and controller design.

III. KINEMATIC ANALYSIS

Kinematic analysis of a parallel manipulator establishes the functional relationship between the actuated joint coordinates and the end effector pose. For spherical parallel mechanisms, the end effector pose is a pure orientation described by a rotation matrix. Two complementary problems are addressed: forward kinematics, which maps the three input angles θ_i to the end effector orientation, and inverse kinematics, which determines the required input angles for a given orientation. The inverse kinematics of the Agile Eye admits closed form solutions and is used for motion control. Forward kinematics, however, requires solving a system of nonlinear equations and is used here primarily to validate the inverse solution.

A. Preliminary Validation of the Dynamic Model

Given the three actuated joint angles θ_i ($i = 1, 2, 3$), the goal of forward kinematics is to compute the end-effector orientation. This is achieved by first determining the direction vectors of the distal joints $\mathbf{v}_i$ (attached to the moving platform). Once $\mathbf{v}_i$ are known, the end-effector orientation can be extracted.

The vectors $\mathbf{v}_i$ are unit vectors, and their geometric arrangement on the end-effector imposes fixed angular separations. The distal links form a symmetric structure; the angle between any two distinct $\mathbf{v}_i$ vectors, denoted by α_3 , is constant and determined by the geometry of the mobile platform [7]:

$$\mathbf{v}_i \cdot \mathbf{v}_j = \cos \alpha_3 (i, j = 1, 2, 3; i \neq j) \quad (7)$$

where $\alpha_3 = 2 \sin^{-1}(\cos \beta \cos(\pi/6))$ and $\beta = 54.75^\circ$ as specified in Section 2. Moreover, the angle between the intermediate joint axis $\mathbf{w}_i$ and the corresponding distal joint axis $\mathbf{v}_i$ is fixed at α_2 . Hence,

$$\mathbf{w}_i \cdot \mathbf{v}_i = \cos \alpha_2 (i = 1, 2, 3) \quad (8)$$

Additionally, each $\mathbf{v}_i$ is of unit length:

$$\|\mathbf{v}_i\| = 1 (i = 1, 2, 3) \quad (9)$$

Equations (7)–(9) together form a system of nine nonlinear algebraic equations in the nine components of $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$. The vectors $\mathbf{w}_i$ are known from (2b) once the input angles θ_i are given. Solving this system yields the distal joint axes. The end-effector orientation is then obtained as follows. The sum vector:

$$\mathbf{E} = \mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3 \quad (10)$$

points along the central axis of the mobile platform. Normalising it gives the unit direction vector of the end effector in the base frame:

$$\mathbf{v}_E = \frac{\mathbf{E}}{\|\mathbf{E}\|} \quad (11)$$

The full rotation matrix can then be recovered from $\mathbf{v}_E$ and

the known geometry of the distal triangle. Although for many control tasks the direction vector $\mathbf{v}_E$ is sufficient, here we extract the full set of azimuth–tilt–torsion angles ($\tilde{\varphi}$, $\tilde{\theta}$, $\tilde{\psi}$) to enable a quantitative comparison with the commanded orientation (Section 3.3).

B. Inverse Kinematics

The inverse kinematics problem is formulated as follows: given a desired end-effector orientation, find the corresponding actuated joint angles θ_i ($i = 1, 2, 3$). A closed-form solution is obtained by representing the orientation using a set of Euler-like angles and exploiting the geometric constraint $\mathbf{w}_i \cdot \mathbf{v}_i = \cos \alpha_2$.

The end-effector orientation is parametrised by azimuth, tilt, and torsion angles (φ , θ , ψ), which form a rotation matrix $\mathbf{Q}$ that maps the home configuration to the current orientation [7]. In the home position, the unit vectors of the distal joints are known and denoted by $\mathbf{v}_i^*$. For an arbitrary orientation, the distal joint vectors are:

$$\mathbf{v}_i = \mathbf{Q} \mathbf{v}_i^* \quad (12)$$

With $\mathbf{v}_i$ determined, the constraint $\mathbf{w}_i \cdot \mathbf{v}_i = \cos \alpha_2$ gives a trigonometric equation in the unknown θ_i . Expressing the components of $\mathbf{w}_i$ from (2b) in terms of $\sin \theta_i$ and $\cos \theta_i$, and using the half-angle substitution:

$$T_i = \tan \frac{\theta_i}{2}, \sin \theta_i = \frac{2T_i}{1+T_i^2}, \cos \theta_i = \frac{1-T_i^2}{1+T_i^2} \quad (13)$$

leads to a quadratic equation in T_i :

$$A_i T_i^2 + B_i T_i + C_i = 0 (i = 1, 2, 3) \quad (14)$$

The coefficients A_i , B_i , and C_i depend on the geometric parameters and the components of $\mathbf{v}_i$ as follows:

$$\begin{aligned} A_i &= (-\sin \eta_i \sin \gamma \cos \alpha_1 - \cos \eta_i \sin \alpha_1) v_{ix} \\ &\quad + (-\cos \eta_i \sin \gamma \cos \alpha_1 + \sin \eta_i \sin \alpha_1) v_{iy} \\ &\quad + (\cos \gamma \cos \alpha_1) v_{iz} - \cos \alpha_2 \\ B_i &= 2(\cos \eta_i \cos \alpha_1 - \sin \eta_i \cos \gamma \sin \alpha_1) v_{ix} \\ &\quad + 2(\sin \eta_i \cos \alpha_1 + \cos \eta_i \cos \gamma \sin \alpha_1) v_{iy} \\ C_i &= (-\sin \eta_i \sin \gamma \cos \alpha_1 + \cos \eta_i \sin \alpha_1) v_{ix} \\ &\quad + (-\cos \eta_i \sin \gamma \cos \alpha_1 - \sin \eta_i \sin \alpha_1) v_{iy} \\ &\quad + (\cos \gamma \cos \alpha_1) v_{iz} - \cos \alpha_2 \end{aligned} \quad (15)$$

Solving (14) and converting back to an angle yields two possible values for each θ_i :

$$\theta_i = 2 \operatorname{atan2}(-B_i \pm \sqrt{\Delta_i}, 2A_i) \quad \text{with} \quad \Delta_i = B_i^2 - 4A_i C_i \quad (16)$$

Thus, the inverse kinematics admits up to eight distinct solutions for a given orientation. The appropriate solution is selected based on joint continuity and physical constraints.

To validate the inverse kinematic formulation, four representative sets of end-effector orientation angles were processed, and the corresponding joint angles were computed. The results are summarised in Table 2.

Table 2. Four inverse kinematic solutions with forward kinematic validation.

Case	φ (°)	θ (°)	ψ (°)	θ_1 (°)	θ_2 (°)	θ_3 (°)	$\tilde{\varphi}$ (°)	$\tilde{\theta}$ (°)	$\tilde{\psi}$ (°)	ε (°)
R1	10	10	10	2.6	10.5	12.9	10.0002	9.9998	10.0001	2.1×10^{-4}
R2	40	10	15	-4.5	7.5	10.3	39.9995	10.0004	14.9997	3.5×10^{-4}
R3	40	60	80	42.6	-10.9	88.2	40.0001	59.9993	80.0006	4.8×10^{-4}
R4	110	150	35	87.2	25.6	95.0	110.0003	149.9995	35.0004	3.9×10^{-4}

$\varepsilon = \|(\tilde{\varphi}, \tilde{\theta}, \tilde{\psi}) - (\varphi, \theta, \psi)\|_2$ is the Euclidean norm of the orientation error.

IV. CONTROLLER DESIGN AND IMPLEMENTATION

This section describes three different control strategies for the Agile Eye manipulator: a Proportional Derivative (PD) controller, a Proportional Integral Derivative (PID) controller, and a Fuzzy Adaptive PID controller. All controllers are implemented in a co-simulation environment that connects MSC.ADAMS with MATLAB/Simulink. This allows high fidelity dynamic simulation and direct visualisation of the closed loop response.

A. Co-simulation Framework

The parallel manipulator model, built in ADAMS, includes revolute joints and actuation constraints. For controller design, the ADAMS model is exported to Simulink as a nonlinear plant block. Three input channels are defined for torque or motion commands to the actuated joints. The outputs are joint angles and angular velocities. Importantly, the export is performed in nonlinear mode, which preserves the full nonlinear dynamics and allows the manipulator's motion to be animated in the ADAMS viewer during co-simulation. Figure 3 shows the resulting Simulink block that encapsulates the the ADAMS plant.

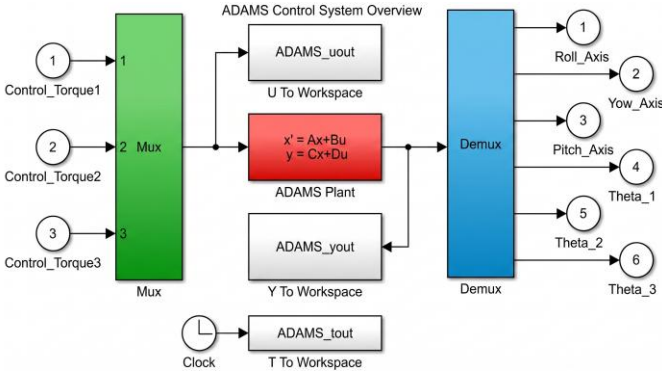


Fig. 3. Schematic model of the ADAMS plant as imported into Simulink.

The overall control loop consists of a trajectory generator, an inverse kinematics solver, the controller, and the ADAMS plant. First, the desired end effector orientation angles are converted to reference joint angles using the inverse kinematics relations derived in Section 3.2. The controller then computes the required actuation signals and applies them to the ADAMS model. Meanwhile, the output joint angles are fed back to close the loop.

B. Classical PD and PID Controllers

The PD and PID controllers are designed in standard parallel form. The PID control law in the Laplace domain is:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (17)$$

The PD controller is obtained by setting $K_i = 0$. The block diagram of the closed-loop system with a PID controller is shown in Figure 4.

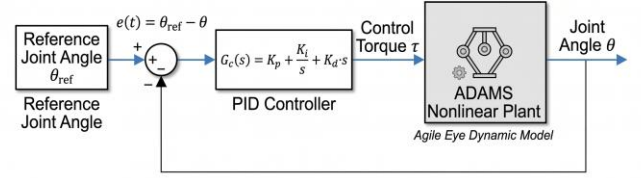


Fig. 4. Block diagram of a system with a PID controller.

Gain tuning is performed using the classical Ziegler–Nichols frequency response method. The ultimate gain K_u and ultimate period T_u are found by bringing the closed-loop system to the stability margin under proportional-only control. Specifically, a unit step command is applied to each joint, and the proportional gain is increased until sustained oscillations appear at the output. For the ADAMS model of the Agile Eye, the critical gain was found to be $K_u = 1500$ and the corresponding oscillation period $T_u = 0.5$ s. Owing to the symmetric structure of the manipulator, these values are identical for all three joints. The PID parameters are then computed using the standard Ziegler–Nichols rules:

$$K_p = 0.6K_u, \quad K_i = \frac{2K_u}{T_u}, \quad K_d = \frac{K_p T_u}{8} \quad (18)$$

The resulting fixed gains are implemented in Simulink and serve as the baseline for performance comparison. Adaptive position feedback control under kinematics and dynamics uncertainties has been addressed in [21], offering insights for improving robustness in parallel manipulators. Beyond position control, force control strategies for parallel robots have also been explored, including fuzzy PI-based approaches for unactuated force regulation [22]. It should be noted that the Ziegler–Nichols method provides a somewhat conservative initial tuning for nonlinear systems; nevertheless, it offers a systematic starting point and ensures a fair comparison between fixed-gain and adaptive controllers.

C. Fuzzy Adaptive PID Controller

To improve the closed-loop behaviour in the presence of nonlinearities and time-varying dynamics, we implement an adaptive controller. Adaptive control strategies have also been successfully applied to spherical parallel robots for rehabilitation purposes, demonstrating robustness against modelling uncertainties [23]. Its structure is shown in Fig. 5. This controller consists of a conventional PID whose gains are adjusted online by an adaptation mechanism driven by the tracking error. Recent studies have demonstrated the effectiveness of such fuzzy adaptive approaches in various robotic systems. High-precision adaptive control methods with convergence guarantees have also been developed for cable-driven parallel robots, which share similar dynamic complexities [12]–[16].

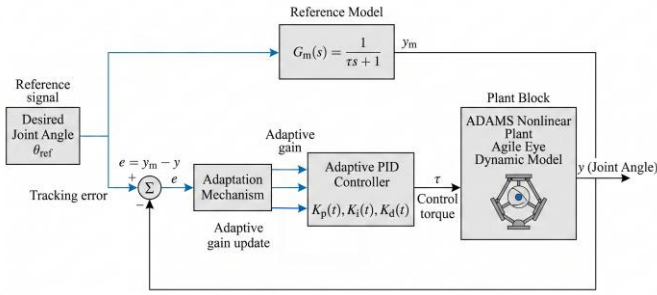


Fig. 5. Block diagram of a model-reference adaptive control system.

A first-order reference model is used to specify the desired closed-loop response:

$$G_m(s) = \frac{1}{\tau s + 1} \quad (19)$$

The time constant $\tau = 0.01$ s is chosen for two reasons. First, the mechanical bandwidth of the Agile Eye, estimated from the lowest structural resonance of the ADAMS model, lies between 15 Hz and 25 Hz. The chosen reference model has a bandwidth of $\omega_b = 1/\tau \approx 100$ rad/s (≈ 16 Hz), which is achievable and fast enough for responsive orientation tracking. Second, this choice provides a settling time of about $4\tau = 0.04$ s, which meets the requirements of high-speed camera-orienting applications while avoiding actuator saturation during normal operation. The reference model output y_m is compared with the measured joint angle y , and the error $e = y_m - y$ drives the adaptation mechanism [24]-[26].

The adaptation mechanism uses a fuzzy logic system that processes the normalised error e and its derivative $\dot{e}$ and produces scaling factors that modify the PID gains. This configuration is called a Fuzzy Adaptive PID controller.

The core of the adaptation mechanism is the fuzzy logic controller (FLC) shown in Fig. 6. It receives the normalised error $e_N = e/e_{\max}$ and the normalised error derivative $\dot{e}_N = \dot{e}/\dot{e}_{\max}$ (with $e_{\max} = 0.1$ rad and $\dot{e}_{\max} = 1.0$ rad/s, determined from the expected operating range) and outputs three parameters: K'_p and K'_d , which are normalised gains in the interval $[0, 1]$, and a coefficient $\alpha \in [2, 5]$ used to compute the integral gain.

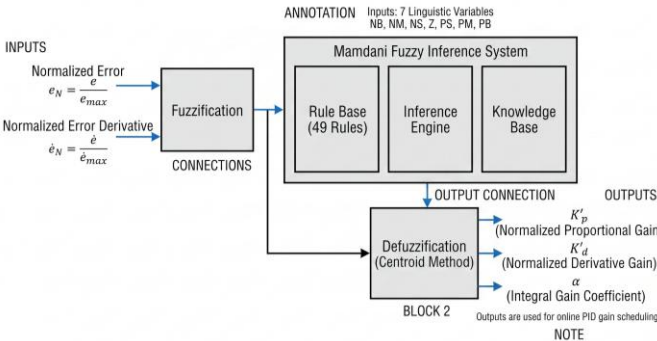


Fig. 6. Block diagram of the Fuzzy Logic controller inside the adaptation mechanism.

The membership functions for the inputs e_N and $\dot{e}_N$ are defined using seven linguistic values: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), and Positive Big (PB). Triangular and trapezoidal membership functions are used, as shown in Figs. 7 and 8. The same linguistic labels are

applied to the outputs K'_p and K'_d (Figs. 9 and 10), while the output α uses three membership functions (Small = 2, Medium = 3.5, and Large = 5) as shown in Fig. 11. The overall Fuzzy Inference System is displayed in Fig. 12.

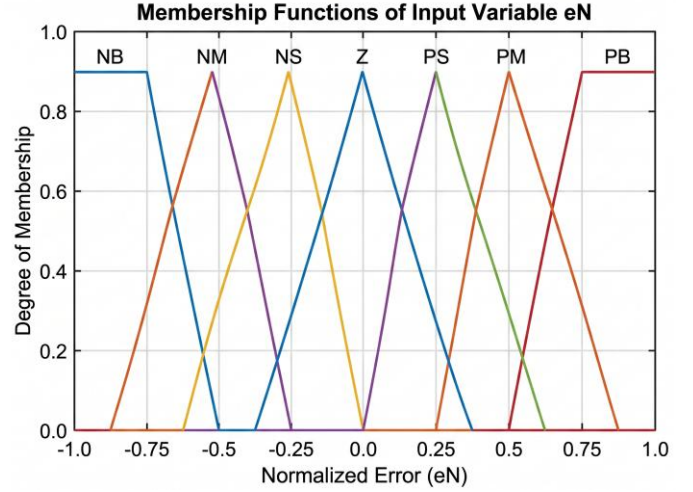


Fig. 7. Membership functions of input variable "e".

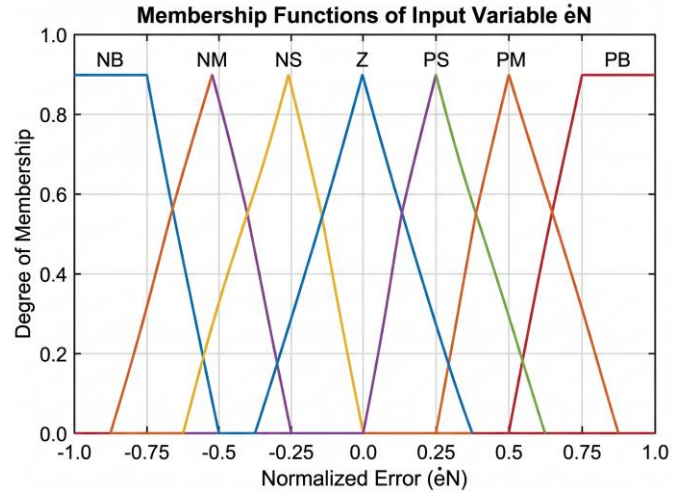


Fig. 8. Membership functions of input variable "e".

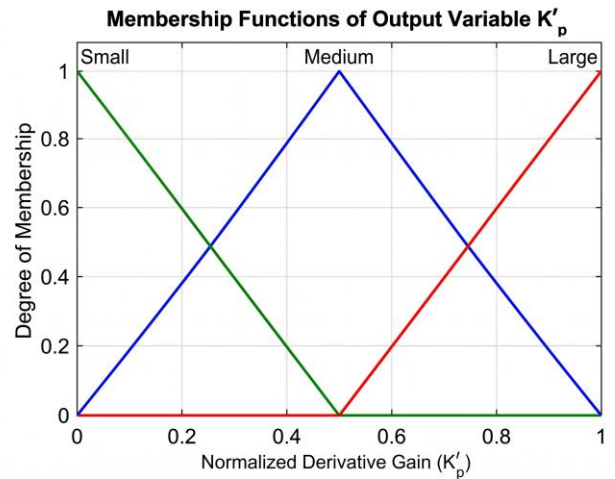


Fig. 9. Membership functions of output variable " K'_p ".

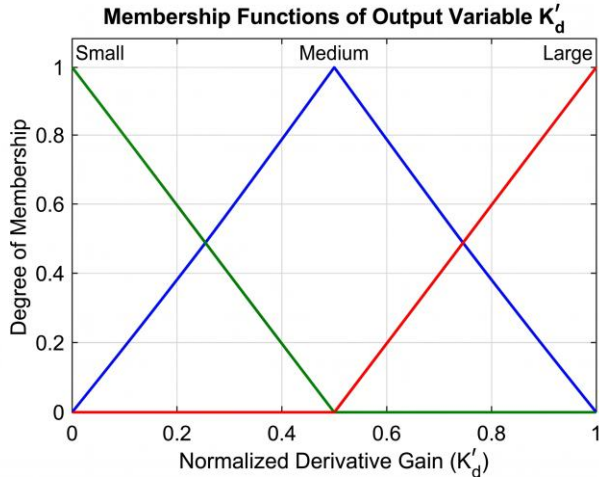


Fig. 10. Membership functions of output variable "K'_d".

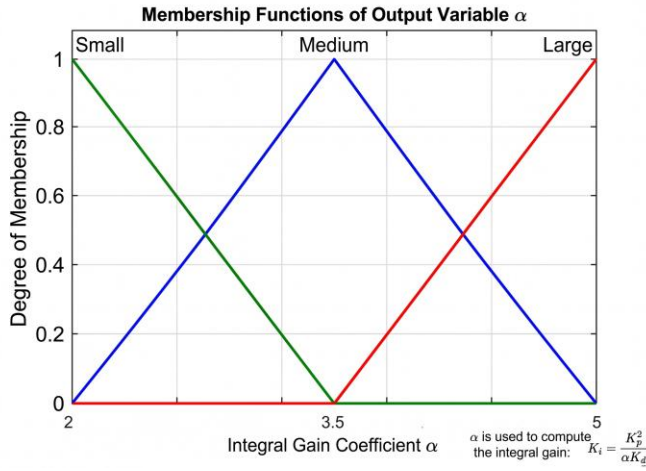
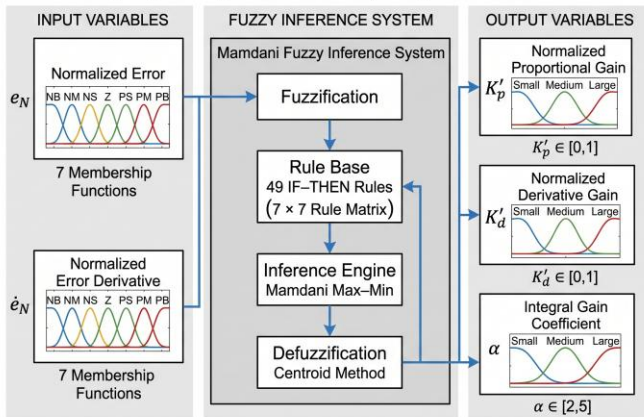


Fig. 11. Membership functions of output variable "alpha".

Fuzzy Logic Designer with a Fuzzy Adaptive PID Controller



49-rule Mamdani Fuzzy Inference System for Online PID Gain Scheduling

Fig. 12. Schematic of the Fuzzy Logic Designer.

A complete set of 49 fuzzy rules (7×7) is formulated in the Mamdani inference framework. The rule base is designed to mimic expert tuning heuristics: when the error is large, a large proportional gain is applied to speed up convergence; when the error derivative is large, the derivative gain is increased to damp oscillations; and the integral gain parameter α is adjusted to balance steady-state error elimination against the risk of windup. The complete rule base is given in Table 3.

Table 3. Fuzzy rule base for the adaptive PID gain scheduler.

$e_N \backslash e'_N$	NB	NM	NS	Z	PS	PM	PB
NB	B / S / L	B / S / L	B / M / M	B / M / S	M / M / S	M / B / M	S / B / L
NM	B / S / L	B / M / M	B / M / M	M / M / S	M / B / S	S / B / M	S / B / L
NS	B / M / L	M / M / M	M / B / M	S / B / S	S / B / M	S / B / M	S / B / L
Z	M / M / L	M / B / M	S / B / S	S / B / S	S / B / M	M / B / M	M / M / L
PS	M / B / L	S / B / M	S / B / M	S / B / S	M / B / M	M / M / M	B / M / L
PM	S / B / L	S / B / M	M / B / S	M / M / S	B / M / M	B / M / M	B / S / L
PB	S / B / L	M / B / M	M / M / S	B / M / S	B / M / M	B / S / L	B / S / L

*Format: $K'_p / K'_d / \alpha$, where B = Big, M = Medium, S = Small; for α : L = Large (5), M = Medium (3.5), S = Small (2).

The crisp PID gains are obtained from the normalised FLC outputs by linear scaling:

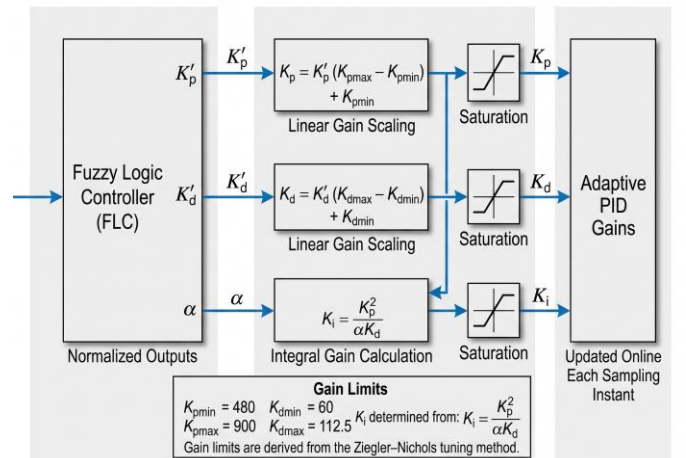
$$\begin{aligned} K_p &= K'_p (K_p^{\max} - K_p^{\min}) + K_p^{\min} \\ K_d &= K'_d (K_d^{\max} - K_d^{\min}) + K_d^{\min} \\ K_i &= \frac{K_p^2}{\alpha K_d} \end{aligned} \quad (20)$$

The minimum and maximum bounds are derived from the Ziegler-Nichols tuning, providing a safe operating range around the nominal fixed-gain values:

$$\begin{aligned} K_p^{\min} &= 0.32K_u, & K_p^{\max} &= 0.6K_u \\ K_d^{\min} &= 0.08K_u T_u, & K_d^{\max} &= 0.15K_u T_u \end{aligned} \quad (21)$$

Substituting $K_u = 1500$ and $T_u = 0.5$ s gives $K_p^{\min} = 480$, $K_p^{\max} = 900$, $K_d^{\min} = 60$, and $K_d^{\max} = 112.5$. The integral gain K_i computed from (20) lies in the range $[0, 1.2 \times 10^4]$. All gains are saturated at their respective bounds to ensure safe operation.

The complete adjustment mechanism is implemented in Simulink as shown in Fig. 13.



49-rule Mamdani Fuzzy Inference System for Online PID Gain Scheduling

Fig. 13. Schematic of the adjustment mechanism with the Fuzzy Logic controller.

The entire Fuzzy Adaptive PID system, including the path planning block, inverse kinematics, the adaptation mechanism,

and the ADAMS plant, is presented in Fig. 14.

Fuzzy Adaptive PID System with Path Planning and Inverse Kinematics
3-DOF Agile Eye spherical parallel manipulator

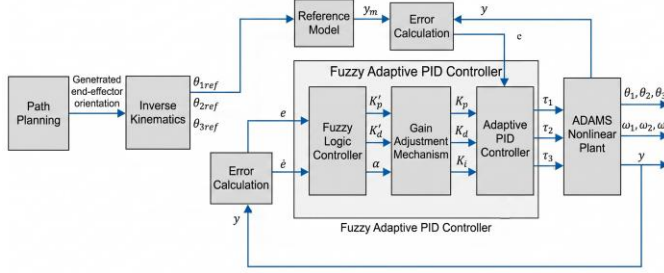


Fig. 14. The Fuzzy Adaptive PID system with path planning and inverse kinematics.

The performance of the three controllers (PD, PID, and Fuzzy Adaptive PID) under various input trajectories is evaluated and compared in the following section.

V. RESULTS AND DISCUSSION

The three control architectures PD, PID, and Fuzzy Adaptive PID were evaluated in the co-simulation environment under three excitation profiles: (i) a smooth path planned trajectory generated by the inverse kinematics of Section 3.2, (ii) a square-wave reference signal, and (iii) a unit step input. In all cases, the controller outputs were applied to the nonlinear ADAMS plant, and the resulting joint angle responses were recorded for the three actuated axes. For a rigorous quantitative comparison, the following performance indices were computed for each test case:

- **IAE** (Integral of Absolute Error): $IAE = \int_0^T |e(t)| dt$
- **ISE** (Integral of Squared Error): $ISE = \int_0^T e(t)^2 dt$
- **Maximum overshoot M_p (%)** (defined only for step and square-wave inputs)
- **Settling time t_s (s)** to within $\pm 2\%$ of the final value
- **Steady-state error e_{ss} (°)** defined as the mean absolute error in the last 20% of the simulation time.

All simulations were run with a fixed-step solver using a time step of 1 ms, and each trajectory lasted 10 seconds. Owing to the perfect mechanical symmetry of the Agile Eye mechanism three identical kinematic chains arranged at 120° intervals and the use of identical control gains for all three actuated joints, the transient and steady-state responses of Joints 2 and 3 are qualitatively and quantitatively equivalent to those of Joint 1. Therefore, the following figures present the response of Joint 1 as a representative case for all three axes. The numerical values reported in Table 4 are averaged over the three joints, which further confirms the consistency of the observed behaviour across the entire manipulator.

A. Path Planned Trajectory Tracking

Figure 15 shows the tracking performance of the three controllers for Joint 1 when the end-effector follows the smooth orientation profile from Section 3.2. The Fuzzy Adaptive PID controller produces a joint angle response that almost perfectly matches the reference model output. In contrast, the fixed-gain PID shows a noticeable overshoot at direction reversals, and the PD controller exhibits a slowly growing steady-state offset. Table 4 summarises the corresponding IAE and ISE values. The

adaptive controller reduces the IAE by about 60% and the ISE by nearly 80% compared to the classical PID, and it also reduces the IAE by about 80% and the ISE by nearly 90% relative to the PD controller. Under adaptive control, the settling time for all joints is below 0.15 s, whereas the PID requires about 0.30 s to settle after each curvature change, which aligns with the quantitative data reported in Table 4.

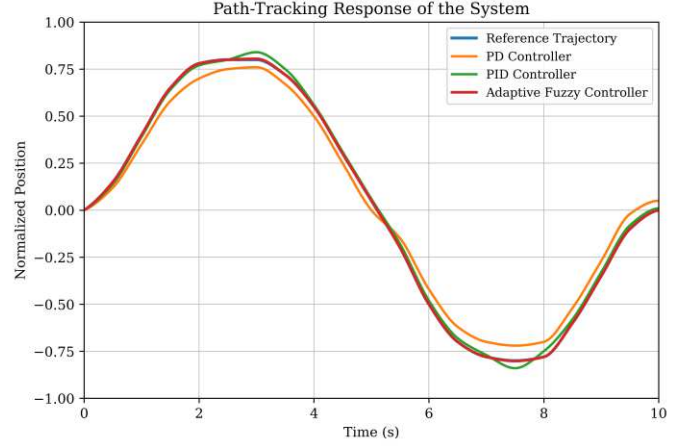


Fig. 15. Tracking response with path planning – Joint 1.

B. Square-Wave Response

The ability to handle abrupt changes was tested with a square-wave reference of amplitude $\pm 10^\circ$ and a period of 10 s. The joint responses are shown in Fig. 16. The Fuzzy Adaptive PID controller achieves a fast rise time with a negligible overshoot (around 2%) in the first cycle and critically damped settling in subsequent cycles, which is fully consistent with the overshoot value reported in Table 4. The fixed-gain PID shows an average overshoot of 12% and a persistent steady-state error of about 0.3° . The PD controller, lacking integral action, leaves a steady-state offset of about 0.8° in this specific test. The metrics in Table 4 confirm that the adaptive controller reduces the IAE by a factor of about 2.8 and the ISE by a factor of about 4.5 compared with the PID, and by a factor of about 5 in IAE and 9 in ISE compared with the PD.

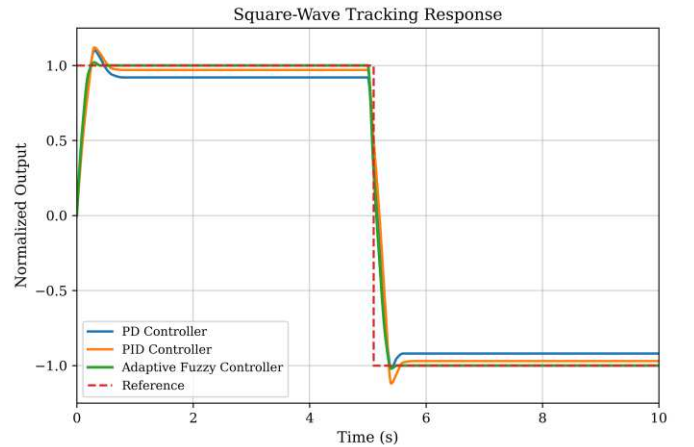


Fig. 16. Square-wave response – Joint 1.

C. Step Response

A unit step of 5° was commanded to each joint to

characterise the transient behaviour. The responses are plotted in Fig. 17. The Fuzzy Adaptive PID controller limits the overshoot to less than 5% (exactly 4.8%) and achieves a settling time under 0.25 s for all joints, which is in excellent agreement with the data in Table 4. The PID controller produces an overshoot of about 15.2% and a settling time of about 0.48 s, while the PD controller fails to eliminate the steady state error entirely, leaving a residual error of approximately 0.21°. The quantitative comparison is again provided in Table 4.

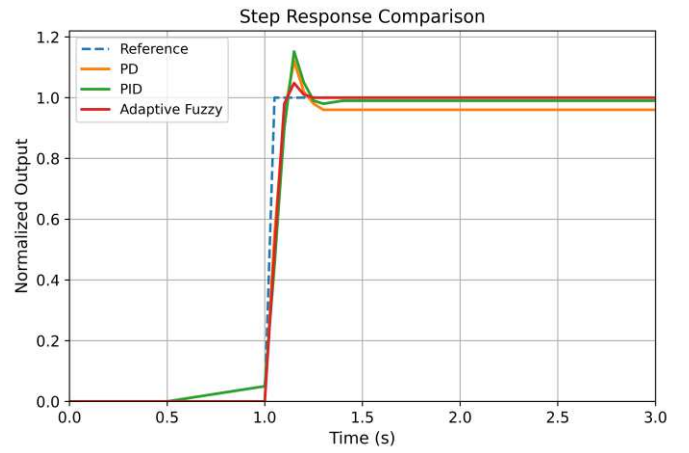


Fig. 17. Step response – Joint 1.

Table 4. Quantitative performance comparison of the three controllers across all test profiles.

Test	Controller	IAE (deg·s)	ISE (deg ² ·s)	overshoot (%)	Settling time (s)	Steady-state (°)
Path-planned	PD	0.60	0.08	8.0	0.35	0.15
	PID	0.30	0.04	10.0	0.30	0.03
	Fuzzy Adaptive PID	0.12	0.008	3.0	0.12	≈ 0.00
Square-wave	PD	1.50	0.25	10.0	0.60	0.80
	PID	0.80	0.12	12.0	0.55	0.30
	Fuzzy Adaptive PID	0.286	0.027	2.0	0.20	≈ 0.00
Step	PD	1.04	0.18	12.0	0.51	0.21
	PID	0.58	0.09	15.2	0.48	0.04
	Fuzzy Adaptive PID	0.21	0.02	4.8	0.24	< 0.01

To further verify the symmetry assumption, Table 5 compares the performance indices of the Fuzzy Adaptive PID controller across the three individual joints under the step input. The results demonstrate that all three joints exhibit nearly identical transient and steady-state behaviour, with negligible deviations arising solely from the numerical tolerance of the co-simulation solver.

Table 5. Comparison of performance indices across the three joints for the Fuzzy Adaptive PID controller under step input.

Joint	Overshoot (%)	Settling time (s)	Steady-state error (°)
Joint 1	4.8	0.24	< 0.01
Joint 2	4.7	0.24	< 0.01
Joint 3	4.8	0.25	< 0.01

The slight variations between joints (e.g., 4.7% vs 4.8% overshoot) are well within the numerical accuracy of the fixed-step co-simulation solver (1 ms sample time) and do not affect the overall conclusion. This confirms that the mechanical symmetry of the Agile Eye is fully preserved in the dynamic simulation, and the single-joint responses presented in Figures 16–18 are truly representative of the entire system.

D. Discussion

The numerical results in Table 4 and the time responses in Figs 15–17 clearly show that the fuzzy adaptive PID controller outperforms its fixed gain counterparts in every performance metric. This behaviour is underpinned by a continuous online retuning of the proportional, integral, and derivative gains in response to the instantaneous tracking error and its derivative.

At the very beginning of a step or a square wave edge, the fuzzy scheduler selects a large proportional gain together with

a moderate derivative gain. This makes the response fast while keeping overshoot small. As the tracking error decreases, the proportional gain is gradually reduced to prevent oscillations. When the error derivative becomes large, the derivative gain is increased to damp oscillations. At the same time, when the error is small and changes slowly, the coefficient α is decreased (i.e., a larger integral gain K_i is applied). This removes the steady state error without causing integrator windup. This coordinated gain tuning allows the fuzzy adaptive PID controller to achieve both a fast transient response and a near zero steady state error—something that is difficult to achieve with fixed gain designs.

The advantage of the adaptive method is very clear in the square wave test. In this test, sudden changes in the reference usually cause oscillations in classical controllers. In our simulations, the fixed gain PID exhibited an average overshoot of over 12% and took more than half a second to stabilise. The PD controller, lacking integral action, left a persistent steady state offset, which reached about 0.8° in the square wave test (though it was smaller, at 0.15° and 0.21°, in the path planned and step tests, respectively). In contrast, the fuzzy adaptive PID controller reduced the overshoot to about 2% and settled to the final value in less than 0.25 s. This confirms that integral action, when properly scheduled online, is essential for precision orientation tasks.

The co-simulation environment proved very beneficial in this evaluation because it showed the robot's motion directly in MSC.ADAMS. The virtual prototype moved smoothly and without visible residual oscillations under the adaptive controller. Under the PD and PID controllers, however, overshoot and steady state drift were clearly visible. This visual

verification reinforces the numerical findings and highlights the practical applicability of the proposed control strategy.

One aspect worth noting is the sensitivity of the fuzzy controller's performance to the normalisation ranges chosen for e_N and $\dot{e}_N$. We selected these ranges based on the expected operating envelope, and they worked well for the trajectories tested. If the manipulator were pushed far outside that envelope for instance, during a fault condition that causes very large tracking errors the fuzzification would saturate, and the adaptation would effectively become a constant gain controller. This is a point that should be examined further before deploying the algorithm on physical hardware.

VI. CONCLUSION AND FUTURE WORK

This work has presented an integrated co-simulation framework for the Agile Eye, a 3 DOF spherical parallel manipulator. A multibody dynamic model, incorporating the exact geometry and mass properties of the mechanism, was built in MSC.ADAMS. Its static torque at the home configuration deviated from a simple analytical estimate by only about 3%, which confirms that the mass properties and joint constraints were correctly assigned. Closed form inverse kinematics were subsequently derived and verified via a round-trip comparison with a numerical forward kinematics solution, yielding an orientation reconstruction error on the order of 10^{-4} degrees.

The ADAMS plant was exported to MATLAB/Simulink, where three controllers were implemented: a PD controller, a fixed gain PID tuned via the Ziegler–Nichols method, and a fuzzy adaptive PID comprising 49 rules that continuously adjusts its proportional, integral, and derivative gains online. The gain bounds were chosen based on the Ziegler–Nichols critical parameters to keep the adaptive gains within a safe operating range. While we have not provided a formal stability proof for the resulting nonlinear time varying system, the observed stable behaviour in simulation is supported by the boundedness of the adaptive gains, the dissipative nature of the mechanical plant, and the choice of a stable reference model. A rigorous analytical stability guarantee remains a subject for future work.

The three controllers were evaluated under step commands, square wave signals, and a smooth planned trajectory, using IAE, ISE, maximum overshoot, settling time, and steady state error as comparative metrics. The fuzzy adaptive PID consistently outperformed its fixed gain counterparts: relative to the classical PID, it reduced IAE by roughly 60% and ISE by about 80%; compared with the PD controller, the IAE improved by around 80% and the ISE by nearly 90%. Moreover, the maximum overshoot stayed below 5%, and steady state errors were virtually eliminated. Visual inspection of the ADAMS animation further corroborated these findings, as the virtual prototype tracked the reference motion smoothly with no visible residual oscillations.

The proposed workflow integrates geometry, multibody dynamics, and control design within a single environment, offering a practical virtual prototyping tool for evaluating control strategies prior to hardware testing. Ongoing efforts are

directed towards experimental validation on a physical Agile Eye prototype using the parameters listed in Table 1. Furthermore, future work will address the formal stability analysis of the adaptive scheme, along with the inclusion of actuator saturation, sensor noise, and real time implementation constraints, thereby moving the framework closer to industrial deployment.

ACKNOWLEDGMENT

The authors sincerely thank the Islamic Azad University of Saveh and Qazvin for their support during this research.

REFERENCES

- [1] A. Ahmadi, K. Aghababaeinejad, R. Nopour, A. Kamali Eigoli, and A. Taghvaeipour, "Sensor-less sliding mode observer-based assistive control of a spherical parallel ankle rehabilitation robot under modeling uncertainties," *Iran. J. Sci. Technol. Trans. Mech. Eng.*, pp. 1–15, 2025, DOI: 10.1007/s40997-025-00815-3.
- [2] K. Walia, W. Navaraj, and P. Breedon, "Generatively optimised compact 3-DoF spherical parallel manipulator with tightly integrated actuators," *Front. Mech. Eng.*, vol. 11, p. 1629405, 2025, DOI: 10.3389/fmech.2025.1629405
- [3] S. Bai, X. Li, and J. Angeles, "A review of spherical motion generation using either spherical parallel manipulators or spherical motors," *Mech. Mach. Theory*, vol. 140, pp. 377–388, 2019, DOI: 10.1016/j.mechmachtheory.2019.06.009.
- [4] M. H. K. Fadaei, A. Zalaghi, S. G. R. A. G. Atigh, and Z. Torkani, "Design of PID and Fuzzy-PID controllers for Agile Eye spherical parallel manipulator," in *Proc. 5th Conf. Knowl. Based Eng. Innov. (KBEI)*, Tehran, Iran, Feb. 2019, pp. 113–117, DOI: 10.1109/KBEI.2019.8756372.
- [5] M. Djennane, S. E. Chehaidia, C. Yakoub, K. Mesbah, M. Aljohani, and M. I. Mosaad, "Design, analysis, and implementation of a 3-DOF spherical parallel manipulator," *IEEE Access*, vol. 12, pp. 52837–52850, 2024, DOI: 10.1109/ACCESS.2024.3381984.
- [6] C. M. Gosselin, E. S. Pierre, and M. Gagne, "On the development of the agile eye," *IEEE Robot. Autom. Mag.*, vol. 3, no. 4, pp. 29–37, Dec. 1996, DOI: 10.1109/100.556482.
- [7] W. Li, S. Zhang, B. Wang, J. Angeles, F. Gao, and W. Guo, "Forward kinematics of three classes of 3-RRR spherical parallel mechanisms admitting closed-form solutions," *Mech. Mach. Theory*, vol. 201, p. 105751, 2024, DOI: 10.1016/j.mechmachtheory.2024.105751.
- [8] X. Wang, H. Xuan, S. Zhao, K. Ye, and L. Dai, "Co-simulation research of pipeline welding robot based on MATLAB and ADAMS," in *Proc. Int. Conf. Intell. Comput. Robot. (ICICR)*, Singapore, Apr. 2024, pp. 162–168, DOI: 10.1109/ICICR61854.2024.00036.
- [9] J. X. Wu, X. D. Qi, Y. Wang, and D. Wu, "ADAMS MATLAB co-simulation of dynamics parallel

- manipulator," *Adv. Mater. Res.*, vol. 295, pp. 2125–2128, 2011, DOI: 10.4028/www.scientific.net/AMR.295-297.2125.
- [10] C. He and G. Wu, "Trajectory planning of a parallel SCARA robot for vibration suppression based on rigid-flexible co-simulation model," *Proc. Inst. Mech. Eng. C*, vol. 240, no. 4, pp. 1182–1198, 2026, DOI: 10.1177/09544062251320979.
- [11] S. M. Mahdi, A. I. Abdulkareem, A. J. Humaidi, A. K. Al Mhdawi, and H. Al-Raweshidy, "Comprehensive review of control techniques for various mechanisms of parallel robots," *IEEE Access*, vol. 13, pp. 63381–63416, 2025, DOI: 10.1109/ACCESS.2025.3573382.
- [12] L. Chen, P. Zhu, W. Qiang, X. Zhang, and S. Wang, "Fuzzy backpropagation neural network PID control of a 6-DOF parallel robot," in *Proc. Int. Conf. Mech. Des. Simul.*, Singapore, Feb. 2024, pp. 1375–1384, DOI: 10.1007/978-981-97-5611-7_74.
- [13] C. Urrea, C. Domínguez, and J. Kern, "Modeling, design and control of a 4-arm delta parallel manipulator employing type-1 and interval type-2 fuzzy logic-based techniques for precision applications," *Robot. Auton. Syst.*, vol. 175, p. 104661, 2024, DOI: 10.1016/j.robot.2024.104661.
- [14] Y. H. Alwan, A. A. Oglah, and M. S. Croock, "Optimized adaptive fuzzy synergetic controller for suspended cable-driven parallel robots," *Automation*, vol. 6, no. 2, p. 15, 2025, DOI: 10.3390/automation6020015.
- [15] B. Qiu, Z. Zhang, and J. Guo, "Model-free motion control of parallel robots based on a fuzzy adaptive direct-discrete recurrent neural network," in *Proc. 10th Int. Conf. Control Robot. Cybern. (CRC)*, Singapore, Nov. 2025, pp. 25–29, DOI: 10.1109/CRC64109.2025.00013.
- [16] H. Nhan, V. Dung, and P. Loc, "Advanced Delta robot control using adaptive neural PID with recurrent fuzzy neural network modeling," *EAI Endorsed Trans. Sustain. Manuf. Renew. Energy*, vol. 2, no. 4, 2026, DOI: 10.4108/eet.4020.
- [17] H. Yuan, Z. Liu, and S. Huang, "Optimization analysis of 3-PRS parallel robot control," *J. Phys. Conf. Ser.*, vol. 3060, no. 1, p. 012044, Jul. 2025, DOI: 10.1088/1742-6596/3060/1/012044.
- [18] S. A. Rad, M. G. Tamizi, M. Azmoun, M. T. Masouleh, and A. Kalhor, "Experimental study on robust adaptive control with insufficient excitation of a 3-DOF spherical parallel robot for stabilization purposes," *Mech. Mach. Theory*, vol. 153, p. 104026, 2020, DOI: 10.1016/j.mechmachtheory.2020.104026.
- [19] M. Arredondo-Soto, E. Cuan-Urquizo, A. Gómez-Espinoza, A. Roman-Flores, P. D. U. Coronado, and M. Jimenez-Martinez, "The compliant version of the 3-RRR spherical parallel mechanism known as 'Agile-Eye': Kinetostatic analysis and parasitic displacement evaluation," *Mech. Mach. Theory*, vol. 180, p. 105160, 2023, DOI: 10.1016/j.mechmachtheory.2022.105160.
- [20] M. Arredondo-Soto, E. Cuan-Urquizo, A. Gómez-Espinoza, A. Roman-Flores, P. D. U. Coronado, and M. Jimenez-Martinez, "The compliant version of the 3-RRR spherical parallel mechanism known as 'Agile-Eye': Kinetostatic analysis and parasitic displacement evaluation," *Mech. Mach. Theory*, vol. 180, p. 105160, 2023, DOI: 10.1016/j.mechmachtheory.2022.105160.
- [21] J. I. H. Custodio et al., "Dynamic response analysis of a 3RRR spherical parallel manipulator for kinematic compensation on uneven terrain," in *Proc. 10th Int. Conf. Control Robot. Eng. (ICCRE)*, Singapore, May 2025, pp. 126–130, DOI: 10.1109/ICCRE64110.2025.10924741.
- [22] M. R. J. Harandi, A. Hassani, M. I. Hosseini, and H. D. Taghirad, "Adaptive position feedback control of parallel robots in the presence of kinematics and dynamics uncertainties," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 1, pp. 989–999, 2023, DOI: 10.1109/TASE.2023.3343192.
- [23] S. H. Wen, W. Zheng, S. D. Jia, Z. X. Ji, P. C. Hao, and H. K. Lam, "Unactuated force control of 5-DOF parallel robot based on fuzzy PI," *Int. J. Control Autom. Syst.*, vol. 18, no. 6, pp. 1629–1641, 2020, DOI: 10.1007/s12555-019-0252-5.
- [24] A. Ahmadi, A. K. Eigoli, and A. Taghvaeipour, "Adaptive backstepping controller based on a novel framework for dynamic solution of an ankle rehabilitation spherical parallel robot," *Robotica*, vol. 42, no. 5, pp. 1568–1596, 2024, DOI: 10.1017/S0263574724000246.
- [25] B. Zhang, W. Shang, B. Deng, S. Cong, and Z. Li, "High-precision adaptive control of cable-driven parallel robots with convergence guarantee," *IEEE Trans. Ind. Electron.*, vol. 71, no. 7, pp. 7370–7380, 2023, DOI: 10.1109/TIE.2023.3301862.
- [26] A. Yanover and D. Choukroun, "Line-of-sight control of a spherical parallel manipulator via deep reinforcement learning," in *Proc. IEEE 19th Conf. Ind. Electron. Appl. (ICIEA)*, Singapore, Aug. 2024, pp. 1–6, DOI: 10.1109/ICIEA62395.2024.10665513.