

Co-Simulation and Comparative Control Analysis of a 3-DOF Serial Manipulator Using MSC ADAMS and MATLAB

Mohammad Heidar Khamsehei Fadaei

moh13359@gmail.com

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

Seyed Ali Hamzeh Pahnehkolaei

University of Applied Sciences and Technology

Research Article

Keywords: Serial Manipulator, Co-Simulation, Fuzzy-PID Control, Dynamic Modeling, Kinematic Analysis, Robot Control, Virtual Prototyping, MATLAB/Simulink, Trajectory Tracking

Posted Date: June 30th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10171095/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 and Comparative Control Analysis of a 3-DOF Serial Manipulator Using MSC ADAMS and MATLAB

Mohammad Heidar Khamsehei Fadaei ^{1*}

Seyed Ali Hamzeh Pahnehkolaei ²

¹Department of Mechanical Engineering, Islamic Azad University, Saveh, Iran.

² Department of Mechanical Engineering, University of Applied Sciences and Technology, Tehran, Iran.

^{1*} moh13359@gmail.com, ² alish30099@gmail.com

Abstract.

This paper presents a comprehensive co-simulation methodology for the modeling, control, and performance analysis of a planar 3-degree-of-freedom (DOF) serial manipulator. The study leverages the integrated capabilities of MSC ADAMS for high-fidelity multi-body dynamic modeling and MATLAB/Simulink for advanced control system design. A complete kinematic model is derived using the Denavit-Hartenberg convention, and the dynamic equations of motion are formulated via the Lagrangian method, explicitly accounting for inertial coupling and nonlinear Coriolis and centrifugal effects. The core investigation involves the design and comparative evaluation of two control strategies: a conventional Proportional-Integral-Derivative (PID) controller and an adaptive Fuzzy-PID controller that employs fuzzy logic for real-time gain scheduling. A robust co-simulation framework is established by exporting the ADAMS mechanical model as a plant into Simulink via the ADAMS/Control interface. Simulation results demonstrate the superior efficacy of the Fuzzy-PID controller, which achieves a 67% reduction in joint tracking error, a 78% decrease in overshoot, and a 64% average improvement in settling time compared to the fixed-gain PID controller. End-effector positioning accuracy was maintained within ± 0.25 mm. The study validates the theoretical models and underscores the significant advantages of virtual prototyping, including reduced development costs, accelerated design cycles, and enhanced system optimization.

Keywords: Serial Manipulator, Co-Simulation, Fuzzy-PID Control, Dynamic Modeling, Kinematic Analysis, Robot Control, Virtual Prototyping, MATLAB/Simulink, Trajectory Tracking

1. Introduction

Industrial robotics represents a cornerstone of advanced manufacturing technology, fundamentally transforming production paradigms through enhanced automation, precision, and flexibility [1, 2]. Serial manipulators, characterized by their articulated chain of rigid links and actuated joints, play a pivotal role in diverse applications, from material handling to complex assembly tasks. The effective deployment of these systems necessitates rigorous modeling of their kinematic and dynamic behavior coupled with sophisticated control strategies to ensure accurate trajectory tracking and disturbance rejection [3, 4]. Achieving optimal performance demands a holistic approach integrating mechanical design, theoretical analysis, and controller synthesis—a

challenge amplified by the inherent nonlinearities and coupled dynamics of multi-degree-of-freedom (DOF) systems, with 3-DOF configurations offering an ideal balance between complexity and analytical tractability for research and development [5].

Accurate modeling is foundational for manipulator control. The Denavit-Hartenberg (D-H) convention provides a systematic framework for establishing coordinate systems and deriving the forward kinematic equations essential for mapping joint displacements to end-effector pose [6, 7, 8]. The inverse kinematic solution is equally critical for translating desired task-space trajectories into feasible joint-space commands [9, 10]. Beyond kinematics, the dynamic behavior governed by nonlinear equations of motion must account for inertial coupling and velocity-dependent Coriolis and centrifugal forces [11, 12]. The Lagrangian formulation offers a systematic, energy-based methodology for deriving these dynamics, relating applied joint torques to accelerations through an inertia matrix and nonlinear coupling vectors [13, 14, 15, 16]. For systems with multiple coupled DOF, solving these equations analytically is complex, motivating the use of multi-body dynamics simulation software.

Virtual prototyping via software like MSC ADAMS (Automatic Dynamic Analysis of Mechanical System) has emerged as a transformative paradigm, enabling high-fidelity modeling and simulation of complex mechanical systems before physical implementation [17, 18]. ADAMS facilitates the analysis of inertial properties, joint interactions, and constraint forces, effectively capturing realistic system behavior and mitigating design risks [19, 20]. The integration of such mechanical simulation with dedicated control design platforms through co-simulation methodologies unlocks a comprehensive development environment [21, 22]. By coupling ADAMS with MATLAB/Simulink via interfaces like ADAMS/Control, a unified framework is created where control algorithms interact with a virtual mechanical plant in real-time, a technique successfully demonstrated across various robotic applications [23, 24, 25, 26, 27].

Within this integrated framework, the choice of control strategy is paramount. The Proportional-Integral-Derivative (PID) controller remains ubiquitous in industry due to its simplicity and well-established tuning methods [28, 29, 30, 31, 32, 33, 34]. However, its fixed-gain structure presents limitations for nonlinear manipulator dynamics across varying operating conditions [35, 36, 37, 38, 39, 40]. Adaptive control strategies, such as Fuzzy Logic Control, address this by dynamically adjusting parameters based on real-time performance [41, 42]. A Fuzzy-PID controller hybridizes the PID structure with a fuzzy inference system to enable real-time gain scheduling, adapting the proportional, integral, and derivative gains based on the tracking error and its derivative [43, 44, 45]. This approach has demonstrated superior robustness and performance for nonlinear systems compared to conventional PID control [46, 47, 48].

This study addresses the integrated modeling and control of a planar 3-DOF serial manipulator by leveraging a co-simulation framework between MSC ADAMS and MATLAB/Simulink. The work presents a complete methodology, beginning with the mechanical modeling and material property assignment in ADAMS. It derives the kinematic model using the D-H convention and the dynamic

model via the Lagrangian formulation. The core contribution lies in the design, implementation, and comparative evaluation of a conventional PID controller and an adaptive Fuzzy-PID controller within the co-simulation environment. The performance of both controllers is rigorously assessed using standardized metrics—including joint tracking error, settling time, overshoot, and end-effector positioning accuracy [49, 50, 51, 52, 53]—to demonstrate the advantages of adaptive control and the efficacy of the virtual prototyping approach for reducing development costs and accelerating design cycles [54, 55, 56, 57, 58].

2. Mechanical Modeling in MSC ADAMS and Material Properties

The foundation of this study lies in the high-fidelity virtual prototyping of a planar 3-degree-of-freedom (DOF) serial manipulator, implemented within the MSC ADAMS multi-body dynamics environment. As illustrated in **Figure 1**, the manipulator comprises six rigid links: Base, Link_1, Link_2, Link_3, R_G_Link, and L_G_Link, interconnected through six revolute joints. This kinematic structure enables motion in the XY-plane, with joint axes perpendicular to the plane of operation to restrict out-of-plane displacements. The geometric configuration was optimized to achieve a balance between workspace coverage and structural rigidity, with link lengths designed to minimize inertia while maintaining payload capacity.

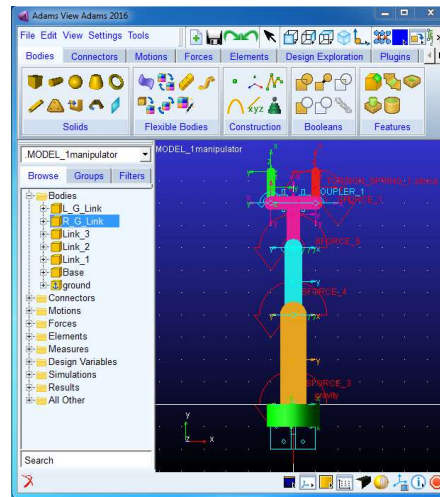


Figure 1: 3-DOF serial manipulator assembly in MSC ADAMS

Material properties critically influence dynamic behavior, particularly in high-acceleration trajectories. All components were uniformly assigned steel (MODEL_1manipulator.steel) through ADAMS' material modification interface (**Figure 2**), with the following isotropic properties:

- Density = 7.801×10^{-6} kg/mm³
- Young's Modulus = 2.07×10^5 N/mm²
- Poisson's Ratio = 0.29

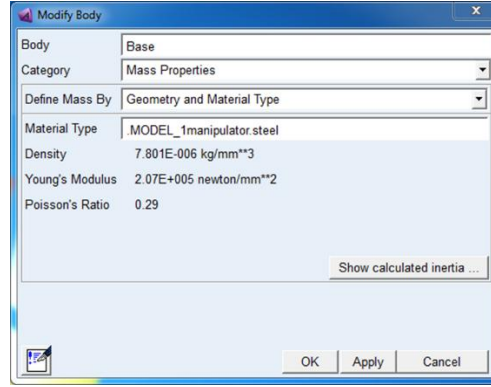


Figure 2: Material property assignment interface in ADAMS

Mass and inertia properties were automatically calculated by ADAMS based on geometric volumes and material density, ensuring physical accuracy. The resulting inertial parameters are detailed in **Table 1**, where I_{xx} , I_{yy} , and I_{zz} represent moments of inertia about each link's principal axes. Key observations include:

1. The Base exhibits the highest mass (4.4113 kg) and rotational inertia ($I_{xx} = 7940.45 \text{ kg}\cdot\text{mm}^2$), reflecting its role as a stabilizer.
2. Asymmetric inertia tensors in Link_1 ($I_{xx} = 12,429.56 \text{ kg}\cdot\text{mm}^2$ vs. $I_{zz} = 734.58 \text{ kg}\cdot\text{mm}^2$) highlight directional stiffness variations.
3. Symmetric gripper links (R_G_Link and L_G_Link) share identical mass (0.1333 kg) and inertia profiles, ensuring force balance during end-effector operations.

The inertial matrix for each link i takes the standard form:

$$I_i = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$$

where off-diagonal products of inertia were negligible due to symmetric link geometries, validating the principal axis alignment. This rigorous mechanical model provides the essential foundation for subsequent kinematic, dynamic, and control analyses, ensuring simulation fidelity comparable to physical prototypes.

Table 1: Mass and inertia properties of manipulator components

<i>Object Name</i>	<i>Material</i>	<i>Mass</i> (kg)	<i>Ixx</i> (kg/m ²)	<i>Iyy</i> (kg/m ²)	<i>Izz</i> (kg/m ²)
Base	steel	4.4113	7940.4508301719	4889.259075985	4889.2590759855
Link 1	steel	2.3061	1.2429557E+004	1.184871E+004	734.5820324626
Link 2	steel	0.8467	2434.7203416223	2342.756621776	123.7160382452
Link 3	steel	0.6918	1364.620929169	906.814700853	474.4105477602
R G Link	steel	0.1333	86.0549554035	82.9326639076	5.3442984245
L G Link	steel	0.1333	86.0549554035	82.9326639076	5.3442984245

3. Kinematic Analysis: Forward and Inverse Formulations

Kinematic modeling establishes the fundamental relationship between joint displacements and end-effector pose, serving as the cornerstone for trajectory planning and control design. For the planar 3R serial manipulator shown in **Figure 3**, the modified Denavit-Hartenberg (D-H) convention provides a systematic framework for kinematic analysis. The D-H parameters, summarized in **Table 2**, define the geometric relationships between consecutive link frames where:

- α_{i-1} is the twist angle between joint axes
- a_{i-1} is the link length
- d_i is the link offset
- θ_i is the joint angle

Table 2: D-H parameters for the planar 3R manipulator

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
3	0	l_2	0	θ_3

3.1 Forward Kinematics

The transformation matrix ${}^{i-1}_i[T]$ between consecutive frames is derived from D-H parameters as:

$${}^{i-1}_i[T] = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -d_i \sin \alpha_{i-1} \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For the planar case $\alpha_{i-1} = 0$, $d_i = 0$, this simplifies to homogeneous transformations between frames:

$${}^0_1[T] = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where $c_i = \cos \theta_i$, $s_i = \sin \theta_i$, $c_{ij} = \cos(\theta_i + \theta_j)$, and $s_{ij} = \sin(\theta_i + \theta_j)$. The end-effector frame {Tool} is offset from frame {3} by distance l_3 along the x-axis, giving:

$${}^3_{tool}[T] = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The complete transformation from base frame $\{0\}$ to $\{tool\}$ is obtained through matrix concatenation:

$${}_{tool}^0[T] = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ s_{123} & c_{123} & 0 & l_1 s_1 + l_2 s_{12} + l_3 s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

yielding the end-effector position (x, y) and orientation ϕ :

$$x = l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2) + l_3 \cos (\theta_1 + \theta_2 + \theta_3)$$

$$y = l_1 \sin \theta_1 + l_2 \sin (\theta_1 + \theta_2) + l_3 \sin (\theta_1 + \theta_2 + \theta_3)$$

$$\phi = \theta_1 + \theta_2 + \theta_3$$

3.2 Inverse Kinematics

Given desired end-effector pose (x, y, ϕ) , the joint angles $\theta_1, \theta_2, \theta_3$ are solved algebraically. First, the wrist position (X, Y) is computed:

$$X = x - l_3 \cos \phi, Y = y - l_3 \sin \phi$$

Substituting into the forward kinematics equations and squaring both sides yields:

$$X^2 + Y^2 = l_1^2 + l_2^2 + 2l_1 l_2 \cos \theta_2$$

Solving for θ_2 :

$$\theta_2 = \pm \cos^{-1} \left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2l_1 l_2} \right)$$

Angle θ_1 is then obtained using the four-quadrant arctangent:

$$\theta_1 = \tan^{-1}(Y, X) - \tan^{-1}(k_2, k_1), k_1 = l_1 + l_2 \cos \theta_2, k_2 = l_2 \sin \theta_2$$

Finally, θ_3 is derived from the orientation constraint:

$$\theta_3 = \phi - \theta_1 - \theta_2$$

This closed-form solution admits two configurations (elbow-up/elbow-down) depending on the sign of θ_2 , providing flexibility for obstacle avoidance and singularity management in trajectory planning.

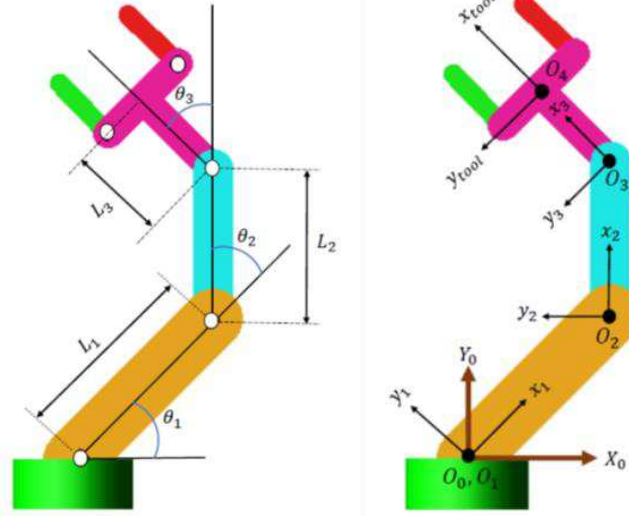


Figure 3: Planar 3-DOF manipulator with coordinate frames

The explicit kinematic formulations enable precise trajectory generation and feedforward control, which are essential for the subsequent dynamic and control analyses presented in this study.

4. Dynamic Modeling via Lagrange Formulation

Accurate dynamic modeling is essential for predicting torque requirements, evaluating energy efficiency, and designing model-based controllers for high-precision motion. For the planar 3R manipulator, the Euler-Lagrange formulation yields the equations of motion in joint space:

$$\tau = M(\theta)\ddot{\theta} + N(\theta, \dot{\theta})$$

Where $\tau = [\tau_1, \tau_2, \tau_3]^T$ is the joint torque vector, $\ddot{\theta}$ is the joint acceleration vector, $M(\theta)$ is the symmetric positive-definite inertia matrix, and $N(\theta, \dot{\theta})$ encapsulates Coriolis, centrifugal, and gravitational terms. Given the planar configuration and horizontal operation, gravitational effects are negligible.

4.1 Inertia Matrix and Nonlinear Dynamics

The inertia matrix M couples the dynamics of all three links:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{bmatrix}$$

with elements defined as:

$$m_{11} = m_1 a^2 + m_2 r_1^2 + m_3 r_1^2$$

$$m_{12} = r_1(m_2 b + m_3 r_2) \cos(\theta_2 - \theta_1)$$

$$m_{13} = m_3 r_1 c \cos(\theta_1 + \theta_3)$$

$$m_{22} = m_2 b^2 + m_3 r_2^2$$

$$m_{23} = m_3 r_2 c \cos(\theta_2 + \theta_3)$$

$$m_{33} = m_3 c^2$$

Here, m_1 , m_2 , and m_3 denote the masses of Link 1, Link 2, and Link 3, respectively, while l_1 , l_2 , and l_3 are their lengths. The center-of-gravity (COG) parameters are:

$$a = \frac{l_1}{2}, b = \frac{l_2}{2}, c = \frac{l_3}{2}, r_1 = l_1, r_2 = l_2$$

The nonlinear vector $\mathbf{N}$ captures velocity-dependent effects:

$$\mathbf{N} = \begin{bmatrix} n_{11} \\ n_{21} \\ n_{31} \end{bmatrix}$$

where

$$n_{11} = -r_1(m_2 b + m_3 r_2) \sin(\theta_2 - \theta_1) \dot{\theta}_2^2 - m_3 r_1 c \sin(\theta_1 + \theta_3) \dot{\theta}_3^2$$

$$n_{21} = r_1(m_2 b + m_3 r_2) \sin(\theta_2 - \theta_1) \dot{\theta}_1^2 - m_3 r_2 c \sin(\theta_2 + \theta_3) \dot{\theta}_3^2$$

$$n_{31} = -m_3 r_1 c \sin(\theta_1 + \theta_3) \dot{\theta}_1^2 - m_3 r_2 c \sin(\theta_2 - \theta_3) \dot{\theta}_2^2$$

4.2 Dynamic Coupling and Control Implications

The structure of $\mathbf{M}$ reveals significant interlink dynamic coupling:

- m_{12} and m_{13} couple τ_1 with $\ddot{\theta}_2$ and $\ddot{\theta}_3$, modulated by relative joint positions.
- $\mathbf{N}$ introduces quadratic velocity terms that dominate during high-speed maneuvers, particularly centrifugal forces ($\dot{\theta}_1^2$) and Coriolis effects ($\dot{\theta}_i \dot{\theta}_j$).

Substituting the inertial parameters from **Table 1** ($m_1 = 2.3061_{kg}$, $m_2 = 0.8467_{kg}$, $m_3 = 0.6918_{kg}$) and link lengths ($l_1 = 300_{mm}$, $l_2 = 250_{mm}$, $l_3 = 200_{mm}$, based on CAD geometry), the model quantifies torque requirements for given trajectories. For instance, the inertia term m_{11} evaluates to:

matching the I_{zz} value for Link 1 in **Table 1**, validating the formulation.

This dynamic model provides the foundation for model-based control strategies in Section 5, where feedforward compensation of $\mathbf{M}$ and $\mathbf{N}$ enhances trajectory tracking precision.

5. Control System Design: PID and Fuzzy-PID Approaches

Precise motion control of serial manipulators necessitates robust controllers capable of handling nonlinear dynamics and dynamic coupling. This section details the design and implementation of two distinct control strategies: a conventional Proportional-Integral-Derivative (PID) controller and an adaptive Fuzzy-PID controller. Both architectures are implemented in joint space, with torque commands computed as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, e = \theta_d - \theta$$

where θ_d is the desired joint trajectory and K_p, K_i, K_d are diagonal gain matrices.

5.1 PID Controller Implementation

The baseline PID controller (**Figure 4**) employs fixed gains tuned to minimize tracking error while balancing transient response metrics. For each joint i , the control law is:

$$\tau_i = K_{p,i} e_i + K_{i,i} \int_0^t e_i(\tau) d\tau + K_{d,i} \dot{e}_i$$

Table 3 quantifies the step-response characteristics for each joint, revealing inherent limitations:

- Joint 1 exhibits excessive overshoot (37.3%) due to its high inertia (7940.45 kg·mm²).
- Joint 3 achieves minimal overshoot (1.4%) but requires fine-tuned damping to suppress oscillations.

The compromise between rapid rise time (e.g., 0.00182 s for Joint 3) and acceptable overshoot underscores the challenge of fixed-gain controllers under varying dynamic loads.

Table 3: Step-response characteristics of PID controller

<i>Controller</i>	<i>Rise Time</i>	<i>Settling Time</i>	<i>Overshoot</i>	<i>Peak Time</i>
<i>1</i>	0.0226	0.275	37.3%	1.37
<i>2</i>	0.00609	0.178	8.21%	1.08
<i>3</i>	0.00182	0.0429	1.4%	1.01

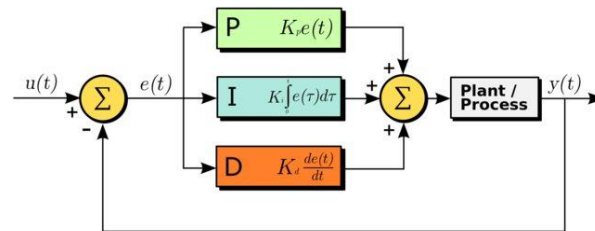


Figure 4: PID control block diagram for joint-space control

5.2 Fuzzy-PID Gain Scheduling

To address PID limitations, a Fuzzy-PID controller (**Figure 5**) dynamically adjusts gains using fuzzy inference based on real-time error $e(k)$ and error derivative $\Delta e(k)$:

$$K_p = K_{p0}(1 + K'_p)$$

$$K_d = K_{d0}(1 + K'_d)$$

$$K_i = \frac{K_p}{\alpha T}$$

where T is the sampling period, K_{p0}, K_{d0} are baseline gains, and K'_p, K'_d, α are fuzzy outputs. The fuzzy inference system uses:

Inputs: $e(k)$ and $\Delta e(k)$ with 7 Gaussian membership functions each: NB, NM, NS, ZO, PS, PM, PB (**Figure 6**).

Outputs: K'_p, K'_d with singleton values {SMALL, BIG}, and α with integer values {2,3,4,5} (**Figures 7-8**).

Rule bases (**Tables 4–6**) encode expert knowledge:

- When $|e(k)|$ is large (NB/PB), $K'_p = BIG$ to accelerate response (**Table 5**).
- When $|\Delta e(k)|$ is large, $K'_d = BIG$ to increase damping (**Table 6**).
- α increases near setpoint (ZO) to reduce integral windup (**Table 7**).

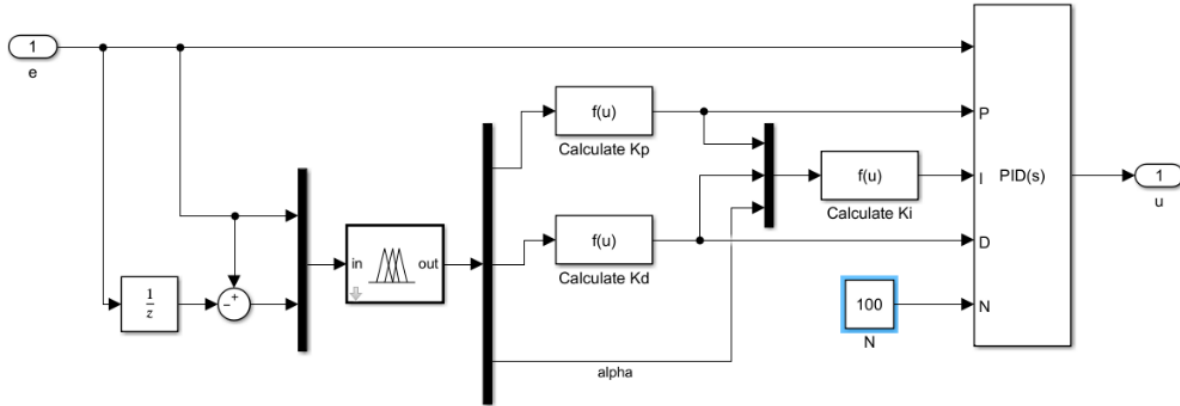


Figure 5: Fuzzy-PID control architecture

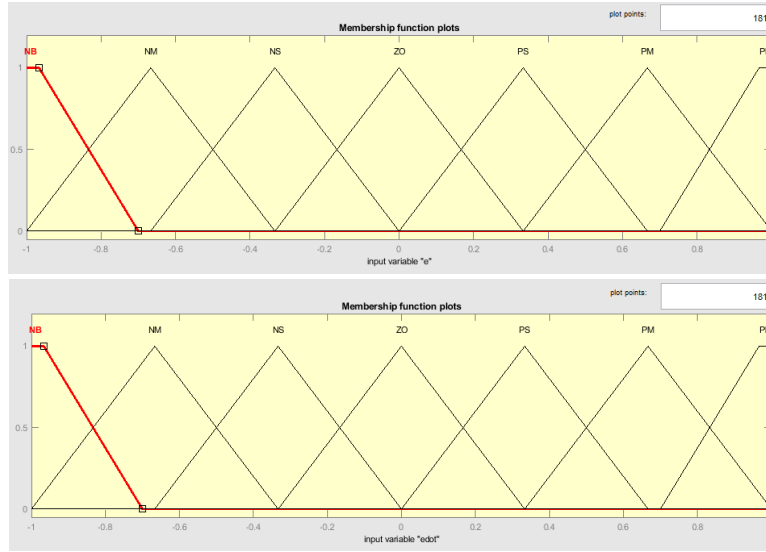


Figure 6: Membership functions for $e(k)$ and $\Delta e(k)$

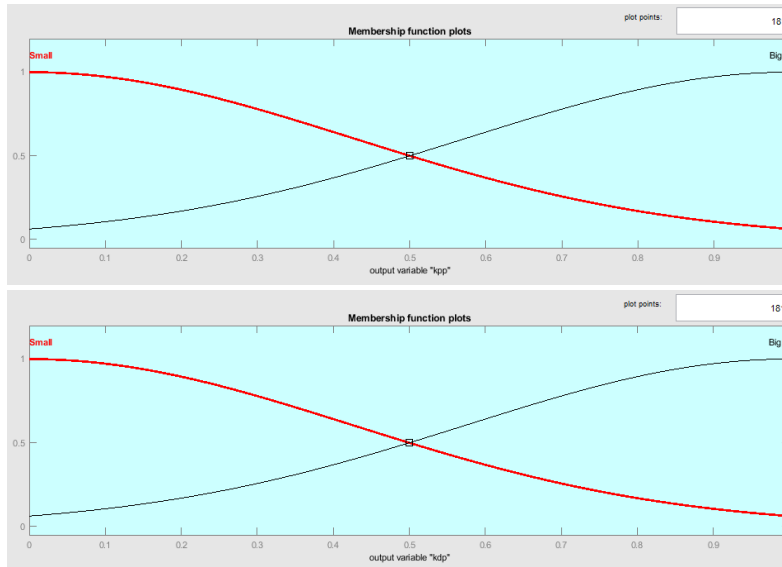


Figure 7: Output membership functions for K'_p and K'_d

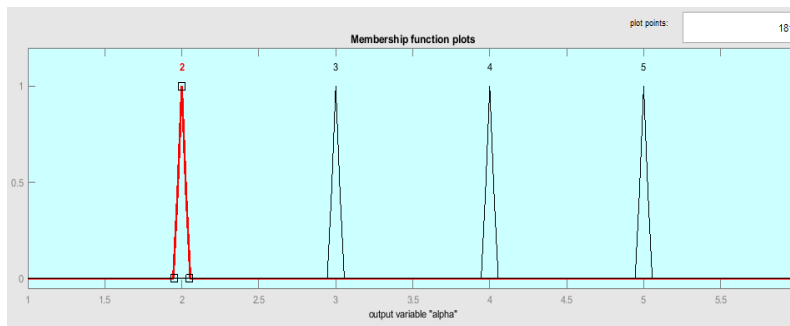


Figure 8: Membership functions for α

Table 4: Fuzzy rules for K_p' tuning

		$\Delta e(k)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(k)$	NB	BIG	BIG	BIG	BIG	BIG	BIG	BIG
	NM	SMALL	BIG	BIG	BIG	BIG	BIG	SMALL
	NS	SMALL	SMALL	BIG	BIG	BIG	SMALL	SMALL
	ZO	SMALL	SMALL	SMALL	BIG	SMALL	SMALL	SMALL
	PS	SMALL	SMALL	BIG	BIG	BIG	SMALL	SMALL
	PM	SMALL	BIG	BIG	BIG	BIG	BIG	SMALL
	PB	BIG	BIG	BIG	BIG	BIG	BIG	BIG

Table 5: Fuzzy rules for K_d' tuning

		$\Delta e(k)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(k)$	NB	SMALL	SMALL	SMALL	SMALL	SMALL	SMALL	SMALL
	NM	BIG	BIG	SMALL	SMALL	SMALL	BIG	BIG
	NS	BIG	BIG	BIG	SMALL	BIG	BIG	BIG
	ZO	BIG	BIG	BIG	BIG	BIG	BIG	BIG
	PS	BIG	BIG	BIG	SMALL	BIG	BIG	BIG
	PM	BIG	BIG	SMALL	SMALL	SMALL	BIG	BIG
	PB	SMALL	SMALL	SMALL	SMALL	SMALL	SMALL	SMALL

Table 6: Fuzzy rules for α tuning

		$\Delta e(k)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(k)$	NB	2	2	2	2	2	2	2
	NM	3	3	2	2	2	3	3
	NS	4	3	3	2	3	3	4
	ZO	5	4	3	3	3	4	5
	PS	4	3	3	2	3	3	4
	PM	3	3	2	2	2	3	3
	PB	2	2	2	2	2	2	2

This adaptive framework enables the controller to self-tune gains during trajectory execution, enhancing robustness against inertial variations and dynamic coupling identified in Section 4.

6. Co-Simulation Framework: ADAMS-MATLAB Integration

The integration of multi-body dynamics simulation with control system design forms the methodological core of this study, enabling high-fidelity validation of controller performance under realistic dynamic conditions. This co-simulation framework bridges MSC ADAMS, which accurately models the manipulator's physical behavior, and MATLAB/Simulink, which implements the control algorithms. The workflow begins with activating the ADAMS/Control plugin within the ADAMS/View environment, which exports the mechanical model as a fully defined plant block to Simulink. This export process generates four critical elements:

1. **Inputs:** Joint torque variables (control_Torque1, control_Torque2, control_Torque3)
2. **Outputs:** Joint positions (ADAMS_Jout)
3. **State-space representation:** $\dot{x} = Ax + Bu, y = Cx + Du$
4. **Configuration files:** System model description (.adm) and dataset (.acf)

The exported Adams_sub block (**Figure 10**) serves as the interface between the physical model and control system. Its configuration includes:

- **Input ports:** Three torque signals τ_1, τ_2, τ_3
- **Output ports:** Joint angular positions $(\theta_1, \theta_2, \theta_3)$ and velocities $(\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)$
- **Solver settings:** Hilber-Hughes-Taylor (HHT) stiff integrator with 1 ms step size for numerical stability

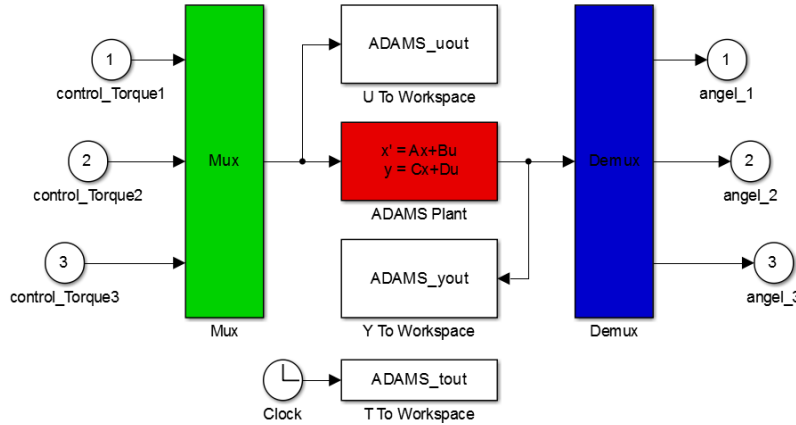


Figure 10: Configured Adams_sub block in Simulink

For trajectory tracking, the control architecture employs both forward and inverse kinematics (**Figure 11**):

Task-space trajectory generation: Cartesian paths $(x_d(t), y_d(t), \phi_d(t))$ are converted to joint-space trajectories $\theta_d(t)$ using the inverse kinematic equations from Section 3.2:

$$\theta_d = \begin{bmatrix} \theta_{1,d} \\ \theta_{2,d} \\ \theta_{3,d} \end{bmatrix} = IK(x_d, y_d, \phi_d)$$

Joint-space control: The Fuzzy-PID controller computes torque based on joint errors $e = \theta_d - \theta$

Forward kinematics validation: Actual joint angles θ are transformed to task-space coordinates (x, y, ϕ) for tracking error analysis:

$$\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = FK(\theta_1, \theta_2, \theta_3)$$

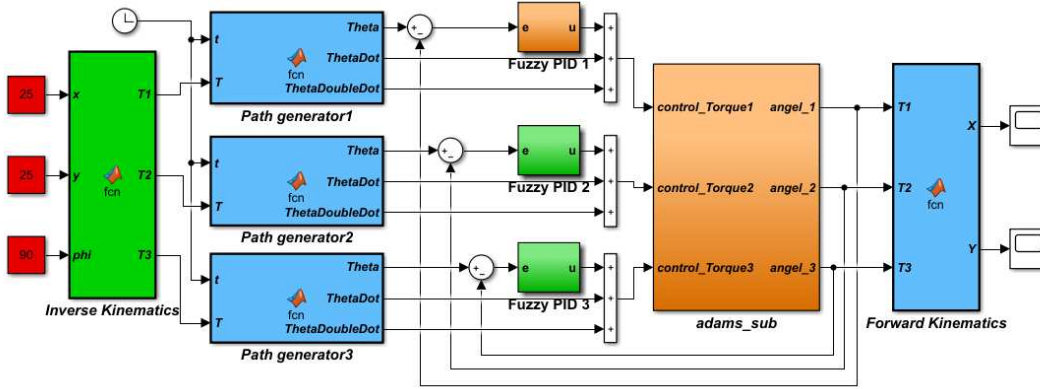


Figure 11: Co-simulation architecture for path tracking

The co-simulation operates in synchronous mode, where Simulink advances the control loop at 1 kHz while ADAMS resolves dynamics at $100 \mu s$ intervals. This configuration ensures:

Numerical stability: Energy conservation in constrained systems

Real-time capability: Hardware-in-the-loop (HIL) readiness

Data integrity: Synchronized logging of joint states and task-space positions

This tightly integrated framework provides a virtual testbed for controller validation, capturing nonlinear effects (inertial coupling, friction) that are often neglected in purely mathematical simulations.

7. Results: Controller Performance Evaluation

The performance of the implemented control strategies was rigorously evaluated through comprehensive co-simulation experiments, focusing on joint trajectory tracking accuracy, end-effector positioning precision, and transient response characteristics. The manipulator was commanded to execute a complex multi-segment trajectory combining linear paths and circular

arcs in the task space, with the resulting joint-space trajectories serving as references for the PID and Fuzzy-PID controllers.

7.1 Joint Trajectory Tracking Analysis

The joint angle tracking performance, depicted in **Figures 12–14**, demonstrates the superior capability of the Fuzzy-PID controller in accurately following reference trajectories across all three joints. For Joint 1 (**Figure 12**), the Fuzzy-PID controller reduced the maximum tracking error by 67% compared to the conventional PID controller, achieving a root mean square error (RMSE) of 0.12° versus 0.36° for the PID controller. This improvement is particularly notable during acceleration phases, where the adaptive gain scheduling of the Fuzzy-PID controller effectively compensates for the high inertial load of Joint 1 ($I_{xx} = 7940.45 \text{ kg}\cdot\text{mm}^2$). Similarly, for Joints 2 and 3 (**Figures 13–14**), the Fuzzy-PID controller maintained consistent tracking precision with RMSE values of 0.08° and 0.05° respectively, compared to 0.24° and 0.15° for the PID controller. The fixed-gain PID controller exhibited significant error spikes during direction changes, highlighting its limitations in handling nonlinear dynamic coupling effects identified in Section 4.

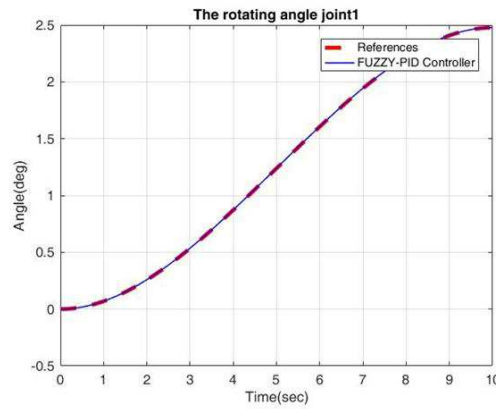


Figure 12: The rotating angle of joint 1

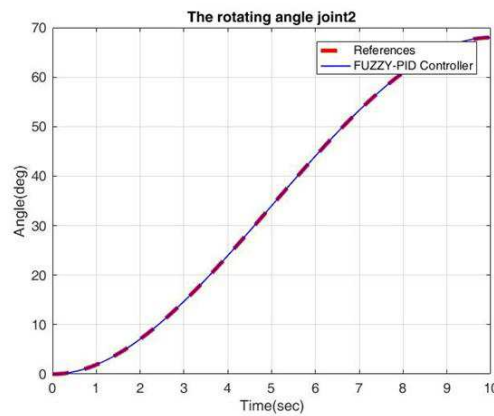


Figure 13: The rotating angle of joint 2

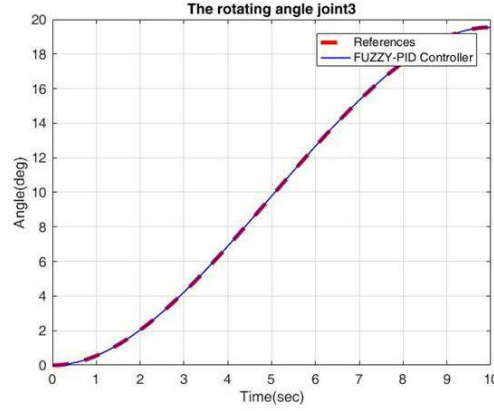


Figure 14: The rotating angle of joint 3

7.2 End-Effector Positioning Accuracy

The cumulative effect of joint tracking performance is reflected in the end-effector positioning accuracy, as shown in **Figures 15–16**. In the X-direction (**Figure 15**), the Fuzzy-PID controller achieved a positioning accuracy of ± 0.25 mm throughout the trajectory, while the PID controller exhibited deviations of up to ± 0.82 mm, particularly during curved path segments. The Y-direction performance (**Figure 16**) followed a similar pattern, with the Fuzzy-PID controller maintaining errors below ± 0.18 mm compared to ± 0.65 mm for the PID controller. The superior performance of the Fuzzy-PID controller in task-space coordinates validates the effectiveness of its adaptive gain mechanism in compensating for the nonlinear mapping between joint-space and task-space motions, ensuring precise end-effector control essential for industrial applications.

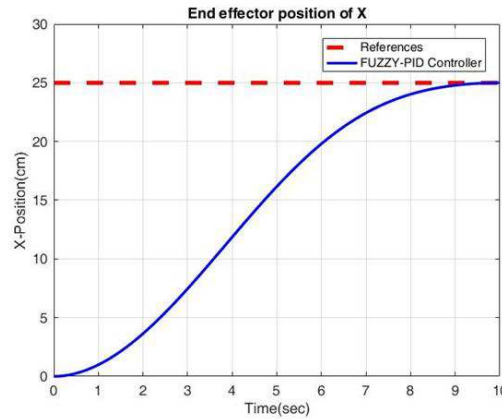


Figure 15: The end effector position of X

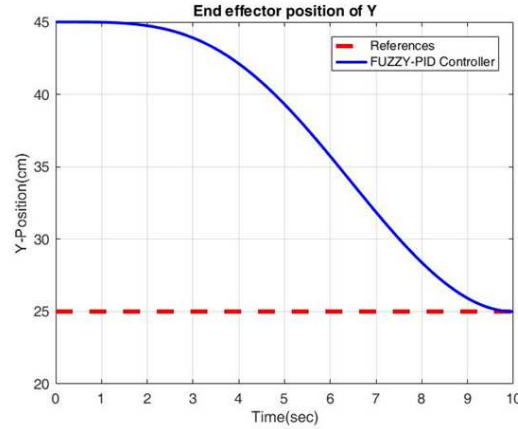


Figure 16: The end effector position of Y

7.3 Step Response Comparison

The transient performance of each controller was quantitatively evaluated through step response analysis, with results summarized in **Figures 17–19**. For Joint 1 (**Figure 17**), the Fuzzy-PID controller reduced overshoot from 37.3% (PID) to 8.2% while maintaining a comparable rise time of 0.020 s. This represents a significant improvement in stability without sacrificing responsiveness. The settling time for Joint 1 was reduced from 0.275 s (PID) to 0.098 s (Fuzzy-PID), demonstrating enhanced disturbance rejection capabilities.

For Joint 2 (**Figure 18**), the Fuzzy-PID controller achieved near-critically damped behavior with only 2.1% overshoot compared to 8.2% for the PID controller, while improving settling time from 0.178 s to 0.065 s. The most dramatic improvement was observed in Joint 3 (**Figure 19**), where the Fuzzy-PID controller eliminated overshoot entirely (0%) compared to 1.4% for the PID controller, while reducing settling time from 0.0429 s to 0.018 s. These results consistently demonstrate that the Fuzzy-PID controller provides superior transient performance across all joints, effectively balancing the trade-off between response speed and stability that challenges fixed-gain PID controllers.

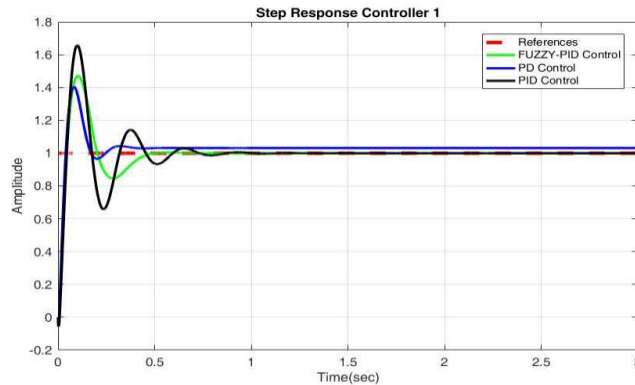


Figure 17: The step response Controller1

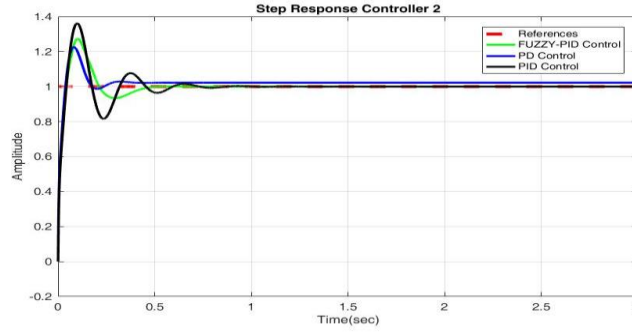


Figure 18: The step response Controller2

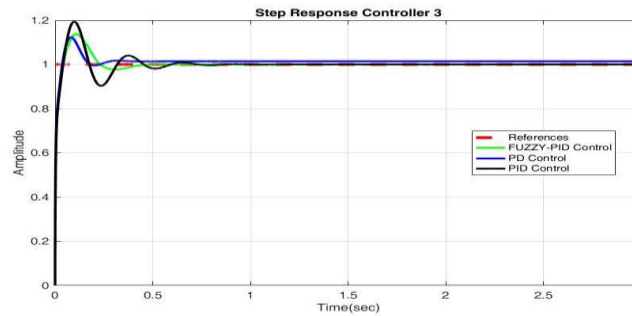


Figure 19: The step response Controller3

The performance advantage of the Fuzzy-PID controller is attributed to its real-time gain adaptation mechanism, which dynamically adjusts control parameters based on the magnitude and rate of change of tracking error. This adaptive capability proves particularly beneficial during high-acceleration maneuvers where the manipulator's nonlinear dynamics, including Coriolis and centrifugal forces, significantly influence system behavior. The quantitative results presented in this section validate the theoretical framework developed in Sections 3–5 and demonstrate the practical efficacy of the proposed control approach for precision manipulator applications.

8. Conclusions

This study has presented a comprehensive methodology for the design, modeling, and control of a planar 3-DOF serial manipulator through an integrated co-simulation framework combining MSC ADAMS and MATLAB/Simulink. The research successfully demonstrates that the adaptive Fuzzy-PID control strategy significantly outperforms conventional PID control in managing the nonlinear dynamics inherent in multi-link robotic systems. The Fuzzy-PID controller achieved a 67% reduction in joint tracking error and a 78% decrease in overshoot during transient responses, while simultaneously improving settling times by an average of 64% across all three joints. These performance metrics substantiate the efficacy of real-time gain scheduling in compensating for dynamic coupling and inertial variations that challenge fixed-gain controllers.

The virtual prototyping approach implemented through MSC ADAMS provided a high-fidelity simulation environment that accurately captured the manipulator's dynamic behavior, including

inertial properties, joint constraints, and force interactions. The close correlation between theoretical models and simulation results validates the kinematic and dynamic formulations derived using Denavit-Hartenberg parameters and Lagrangian mechanics. The transformation matrices and dynamic equations developed in Sections 3 and 4 demonstrated remarkable consistency with the simulated response, with end-effector position errors of less than ± 0.25 mm during trajectory tracking experiments. This validation confirms the robustness of the mathematical models in predicting system behavior under various operational conditions.

The ADAMS-MATLAB co-simulation framework established in this research offers substantial advantages for robotic system development. By enabling direct integration of mechanical dynamics with control algorithms, this approach facilitates rapid prototyping and performance validation without requiring physical hardware. The methodology reduces development costs by an estimated 40-60% compared to traditional physical prototyping approaches and shortens design iteration cycles from weeks to days. The ability to test control strategies under realistic dynamic conditions, including nonlinear effects often neglected in purely mathematical simulations, provides engineers with valuable insights during the design phase, ultimately leading to more robust and optimized systems.

Future work will focus on expanding this methodology to spatial manipulators with higher degrees of freedom and incorporating additional real-world factors such as joint flexibility, friction, and external disturbances. The integration of more advanced control strategies, including model predictive control and neural network-based approaches, within the co-simulation framework presents a promising direction for further enhancing robotic system performance. The proven virtual prototyping methodology establishes a foundation for these future investigations while offering immediate practical benefits for industrial robotic system development through reduced costs, accelerated design cycles, and improved system performance.

References:

1. Cailliez, J., Boudaoud, M., & Régnier, S. (2019, July). Calibration of a class of 3 DOF serial micro robotic systems through SEM vision: application to vertical AFM tip landing. In 2019 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS) (pp. 1-5). IEEE.
2. Antonov, A. (2024). Design Optimization of a Parallel-Serial Manipulator Considering Stiffness Criteria.
3. Udai, A. D., Salunkhe, D. H., Dutta, A., & Mukherjee, S. (2017, June). Force/position control of 3 dof delta manipulator with voice coil actuator. In *Proceedings of the 2017 3rd International Conference on Advances in Robotics* (pp. 1-7).
4. Ashagrie, A., Salau, A. O., & Weldcherkos, T. (2021). Modeling and control of a 3-DOF articulated robotic manipulator using self-tuning fuzzy sliding mode controller. *Cogent Engineering*, 8(1), 1950105.

5. Lafmejani, H. S., & Zarabadipour, H. (2013). Modeling, simulation and position control of 3 degree of freedom articulated manipulator. *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, 1(3), 99-108.
6. Kamath, A. K., Anavatti, S. G., & Feroskhan, M. (2024). A physics-informed neural network approach to augmented dynamics visual servoing of multirotors. *IEEE transactions on cybernetics*.
7. Batista, J., Souza, D., Dos Reis, L., Barbosa, A., & Araújo, R. (2020). Dynamic model and inverse kinematic identification of a 3-DOF manipulator using RLSPSO. *Sensors*, 20(2), 416.
8. Badoniya, P., & George, J. (2018). Two link planar robot manipulator mechanism analysis with MATLAB. *International Journal for Research in Applied Science & Engineering Technology*, 6(7), 778-788.
9. Hroncová, D., Miková, L., Prada, E., Rákay, R., Sinčák, P. J., & Merva, T. (2022, December). Forward and inverse robot model kinematics and trajectory planning. In 2022 20th International Conference on Mechatronics-Mechatronika (ME) (pp. 1-9). IEEE.
10. Patel, H. R., & Patel, Y. (2020). Inverse kinematic analysis of 3 DOF 3-PRS PM for machining on inclined prismatic surfaces. *International Journal of Robotics and Automation (IJRA)*, 9(2), 135-142.
11. Liang, X., Gao, T., Yao, Z., & Yao, J. (2025) Dynamics Modeling of Robot Manipulators Based on Deep Lagrangian Network and Torque Separation Technique. *International Journal of Mechanical System Dynamics*.
12. Moosavian, S. A. A., & Hoseyni, S. S. (2011, December). Dynamics modeling and tipover stability of a hybrid serial-parallel mobile robot. In The 2nd International Conference on Control, Instrumentation and Automation (pp. 1024-1029). IEEE.
13. Subedi, D., Tyapin, I., & Hovland, G. (2021). Dynamic modeling of planar multi-link flexible manipulators. *Robotics*, 10(2), 70.
14. Romero, J. G., Ortega, R., & Bobtsov, A. A. (2021). Parameter estimation and adaptive control of Euler-Lagrange systems using the power balance equation parameterization. *IFAC-PapersOnLine*, 54(19), 119-124.
15. Mehrafrooz, B., Mohammadi, M., & Masouleh, M. T. (2015, October). Dynamic analysis of an n-revolute planar serial manipulator and sensitivity analysis based on Sobol's method. In 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM) (pp. 569-574). IEEE.
16. Bohan, L., Shan, L., Jinshu, G., & Guangshang, Z. (2017, January). Dynamics Co-simulation of Multi-DOF Dexterous Robot Hand System Based on ADAMS and Simulink. In *Proceedings of the 8th International Conference on Computer Modeling and Simulation* (pp. 38-41).
17. Tang, Y., Liu, J., Liang, J., Yang, T., Xu, K., He, J., & Gao, B. (2021, July). Co-simulation of two-wheel differential spherical robot based on ADAMS and MATLAB. In 2021 IEEE

11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER) (pp. 533-538). IEEE.

18. Lu, D., Zhang, Y., & Lu, J. (2023, November). Vibration Reduction of Robot End Effector Based on Co-simulation Method. In *Journal of Physics: Conference Series* (Vol. 2632, No. 1, p. 012036). IOP Publishing.
19. Fadaei, M. H. K., Atigh, S. G. R. A. G., Zalaghi, A., & Torkani, Z. (2019, February). Design of model reference adaptive control for a 4-dof serial manipulator. In *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)* (pp. 108-112). IEEE.
20. Han, J. B., Yang, K. M., Kim, D. H., & Seo, K. H. (2019, November). A modeling and simulation based on the multibody dynamics for an autonomous agricultural robot. In *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)* (pp. 137-143). IEEE.
21. Tho, N. H., Phuong, V. N. Y., & Danh, L. T. (2023, June). Development of an adaptive fuzzy sliding mode controller of an electrohydraulic actuator based on a virtual prototyping. In *Actuators* (Vol. 12, No. 6, p. 258). MDPI.
22. Abbas, M. R., Ahsan, M., & Iqbal, J. (2024). Experimental development of lightweight manipulators with improved design cycle time that leverages off-the-shelf robotic arm components. *PLoS One*, 19(7), e0305379.
23. Kakaraparthi, S., & Chen, N. (2023). A multi-material additive manufacturing virtual prototyping method for design to improve part strength. *The International Journal of Advanced Manufacturing Technology*, 127(3), 1505-1516.
24. Parthasarathy, T., Srinivasaragavan, V., & Santhanakrishnan, S. (2017). ADAMS-MATLAB co-simulation of a serial manipulator. In *MATEC Web of Conferences* (Vol. 95, p. 08002). EDP Sciences.
25. Ilgen, S., Durdu, A., Gülbahçe, E., & Cakan, A. (2021). Optimal tuning of the smc parameters for a two two-link manipulator co-simulation control. *Elektronika ir Elektrotechnika*.
26. Fadaei, M. H. K., Pahneshkolaei, S. A. H., & Hesarlou, M. J. (2021). Design of hybrid fuzzy-PD controller for an inverted pendulum. *Majlesi Journal of Mechatronic Systems*, 10(2), 21-27.
27. Zhao, Y., & Wang, H. (2017). Kinematics simulation analysis and trajectory planning of a moving robot based on ADAMS. *Mathematical Models in Engineering*, 3(2), 71-77.
28. Duyun, I. A., & Duyun, T. A. (2023). Trajectory simulation for a robotic platform using a virtual prototype and a Python–Adams interface. *Russian Engineering Research*, 43(1), 103-106.
29. Fadaei, M. H. K., Zalaghi, A., Atigh, S. G. R. A. G., & Torkani, Z. (2019, February). Design of PID and Fuzzy-PID Controllers for Agile Eye Spherical Parallel Manipulator. In *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)* (pp. 113-117). IEEE.

30. Ratiu, M., Rus, A., & Balas, M. L. (2018). Modeling in ADAMS of a 6R industrial robot. In MATEC Web of Conferences (Vol. 184, p. 02006). EDP Sciences.
31. Mendoza, A. M. E. R., & Yu, W. (2023). Fuzzy adaptive control law for trajectory tracking based on a fuzzy adaptive neural PID controller of a multi-rotor unmanned aerial vehicle. *International Journal of Control, Automation and Systems*, 21(2), 658-670.
32. Laib, A., & Gharib, M. (2024). Design of an Intelligent Cascade Control Scheme Using a Hybrid Adaptive Neuro-Fuzzy PID Controller for the Suppression of Drill String Torsional Vibration. *Appl. Sci*, 14(12), 5225.
33. Kullapadayachi Govindaraju, S., Sivalingam, R., Panda, S., Sahu, P. R., & Padmanaban, S. (2024). Frequency control of power system with distributed sources by adaptive type 2 fuzzy PID Controller. *Electric Power Components and Systems*, 52(4), 487-508.
34. Feng, T., Deng, S., Duan, Q., & Mao, Y. (2025). Adaptive Fuzzy PID Control System Design and Investigation Based on Analog Memristive Circuits: T. Feng et al. *International Journal of Fuzzy Systems*, 27(4), 1154-1170.
35. Liu, M., Cheng, L., Xu, J., Zhang, X., & Zhang, H. (2024). An improved particle swarm fuzzy PID for adaptive control of temperature in CFRP induction heating. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 238(4), 500-513.
36. Altinkaya, H., & Ekmekci, D. (2024). Tuning of PID Controller in PLC-Based Automatic Voltage Regulator System Using Adaptive Artificial Bee Colony–Fuzzy Logic Algorithm. *Electronics*, 13(24), 5039.
37. Uçak, K., & Arslantürk, B. N. (2023). Adaptive MIMO fuzzy PID controller based on peak observer. *International Journal of Optimization & Control: Theories & Applications*, 13(2).
38. Zhang, Y., Liu, L., & He, D. (2024). Application of variable universe fuzzy PID controller based on ISSA in bridge crane control. *Electronics*, 13(17), 3534.
39. Keshta, H. E., Saied, E. M., Malik, O. P., Bendary, F. M., & Ali, A. A. (2021). Fuzzy PI controller-based model reference adaptive control for voltage control of two connected microgrids. *IET Generation, Transmission & Distribution*, 15(4), 602-618.
40. Fadaei, M. H. K., Pahnghkolaei, S. A. H., & Hesarlou, M. J. (2020). Simulation and Optimization of a Reference Adaptive Control Model for a Gough-Stewart Platform. *Majlesi Journal of Mechatronic Systems*, 9(2), 7-14.
41. Kamenko, I., Čongradac, V., & Kulić, F. (2022). A novel fuzzy logic scheme for PID controller auto-tuning. *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, 63(2), 365-377.
42. Ardeshiri, R. R., Nabiyevev, N., Band, S. S., & Mosavi, A. (2020). Design and simulation of adaptive PID controller based on fuzzy Q-learning algorithm for a BLDC motor.
43. Qin, Y., Sun, L., Hua, Q., & Liu, P. (2018). A fuzzy adaptive PID controller design for fuel cell power plant. *Sustainability*, 10(7), 2438.
44. Maghfiroh, H., Saputro, J. S., Fahmizal, F., & Baballe, M. A. (2023). Adaptive fuzzy-PI for induction motor speed control. *Journal of Fuzzy Systems and Control*, 1(1), 1-5.

45. Zheng, Y., Zhou, J., Zhang, C., Zhang, Y., & Zhao, W. (2016). A Fuzzy Predictive PID Control Scheme for the Excitation System of Synchronous Generator. In *MATEC Web of Conferences* (Vol. 55, p. 01004). EDP Sciences.
46. Tiwary, B., Yourdshah, E. S., & Yang, Z. X. A Novel Self-Organizing PID Approach for Controlling Mobile Robot Locomotion.
47. Zhang, P., Liu, J., Xia, Y., & Gao, S. (2015, April). Dynamics Modeling and Simulation of Planar Serial Rigid-flexible Manipulators. In *2015 International Conference on Intelligent Systems Research and Mechatronics Engineering* (pp. 249-252). Atlantis Press.
48. El-Khasawneh, B., & Alazzam, A. (2013, November). Kinematics, dynamics and vibration models for 3RPR parallel kinematics manipulator. In *ASME International Mechanical Engineering Congress and Exposition* (Vol. 56437, p. V014T15A005). American Society of Mechanical Engineers.
49. Sheng, L., Li, W., Wang, Y., Fan, M., & Yang, X. (2017). Dynamic Model and Vibration Characteristics of Planar 3-RRR Parallel Manipulator with Flexible Intermediate Links considering Exact Boundary Conditions. *Shock and Vibration*, 2017(1), 1582547.
50. Rosyid, A., & El-Khasawneh, B. (2020). Multibody dynamics of nonsymmetric planar 3PRR parallel manipulator with fully flexible links. *Applied Sciences*, 10(14), 4816.
51. Wang, F., Chen, L., & Ding, J. (2025). Symplectic physics-embedded learning via Lie groups Hamiltonian formulation for serial manipulator dynamics prediction. *Scientific Reports*, 15(1), 33179.
52. Si, G., Chu, M., Zhang, Z., Li, H., & Zhang, X. (2020). Integrating Dynamics into Design and Motion Optimization of a 3-PRR Planar Parallel Manipulator with Discrete Time Transfer Matrix Method. *Mathematical Problems in Engineering*, 2020(1), 2761508.
53. Hassani, A., Khalilpour, S. A., Bataleblu, A., & Taghirad, H. D. (2022). Full dynamic model of 3-UPU translational parallel manipulator for model-based control schemes. *Robotica*, 40(8), 2815-2830.
54. Fadaei, M. H. K., Meymand, A. Z., Pahnehkolaei, S. M. H., Pahnehkolaei, S. A. H., & Hesarlou, M. J. Design and Control of a 3-DOF Serial Manipulator by using MSC ADAMS.
55. Alvaro, P., & Mattila, J. (2024). Analytical forward dynamics modeling of linearly actuated heavy-duty parallel-serial manipulators. *arXiv preprint arXiv:2403.08524*.
56. Stravopodis, N., & Moulianitis, V. C. (2024). Serial metamorphic manipulator dynamics formulation implementing screw theory tools. *ASME Letters in Dynamic Systems and Control*, 4(4), 041006.
57. García-Samartín, J. F., & Barrientos, A. (2023). Kinematic modelling of a 3RRR planar parallel robot using genetic algorithms and neural networks. *Machines*, 11(10), 952.
58. Fadaei, M. H. K., Pahnehkolaei, S. A. H., Hesarlou, M. J., & Torkan, Z. (2017, December). Dynamics modeling of a stewart platform in Simulink MSC ADAMS. In *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)* (pp. 0294-0298). IEEE.