Optimal PID Control for Quadruped Robot using Puma Optimizer: A Numerical Study

Suvansh Gupta[1], Chahek Sarawagi[1], Jay Dhamija[1], Jyotindra Narayan[2,3], Ashish Singla[1], Achraf Jabeur Telmoudi[4]
[1]*Department of Mechanical Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India*
[2]*Department of Mechanical Engineering, Indian Institute of Technology Patna, Bihar, India*
[3]*Department of Computing, Imperial College London, UK*
[4]*LISIER Laboratory, Higher National Engineering School of Tunis, University of Tunis, Tunisia*
suvanshg0@gmail.com, sarawagi.chahek@gmail.com, jaydhamija@gmail.com
n.jyotindra@gmail.com, ashish.singla@thapar.edu, achraf.telmoudi@ieee.org

***Abstract*—This paper proposes a novel Puma Optimizer-based PID control (PO-PID) strategy for quadruped robots and presents a comprehensive comparison against two established approaches: Harris Hawks Optimization-PID (HHO-PID) and Slime Mould Algorithm-PID (SMA-PID). While conventional PID tuning methods struggle with high-dimensional search spaces and local optima, the PO algorithm leverages adaptive exploration-exploitation mechanisms inspired by puma hunting behavior to achieve globally optimal PID gains. Detailed simulations evaluate the three controllers in terms of trajectory accuracy and control effort. The PO-PID controller achieved the lowest joint trajectory error and significantly outperformed HHO-PID and SMA-PID in Cartesian accuracy while maintaining a lower or comparable torque demand. Additionally, PO-PID demonstrated faster convergence and more consistent optimization, indicating its robustness and reliability. These findings validate the PO-PID controller as a powerful and efficient solution for enhancing quadruped robot locomotion.**

***Index Terms*—Puma Optimizer, PID control, quadruped robots, trajectory tracking, HHO-PID, SMA-PID, control effort.**

I. Introduction

Quadruped robots have been the subject of extensive research due to their remarkable stability and load-bearing capacity, which are comparatively less complex than those of hexapod and octopod robots [1]. These inherent advantages have facilitated the integration of quadruped robots into a diverse range of operational environments, accommodating both rugged and flat terrain [2]. In contrast, wheeled robots can achieve high speeds exclusively on relatively smooth surfaces [3]; however, they demonstrate reduced mobility in more uneven landscapes than legged robots [4]. Quadruped robots have the advantages of strong obstacle capability, less energy consumption, high flexibility, and good stability on uneven and rough terrain.

The performance of these quadruped robots largely depends on their control systems, which must balance stability, versatility, and energy efficiency across various conditions. Quadruped robots often require a range of controllers, including MPC, LQR, and adaptive control, to handle complex locomotion tasks [5], [6]. These advanced strategies, while effective, can be computationally intensive, require accurate system modeling, or face challenges in real-time adaptability. Therefore, PID controllers remain relevant due to their simplicity, efficiency, and compatibility with existing systems [7].

Optimal PID control for robotic manipulators has been a focus of research to improve performance and stability in real-time scenarios [8]. Rashad et al. compared PID and Non-linear PID controllers for master-slave robotic systems, using genetic algorithms to obtain optimal gains [9]. Recent algorithms such as the Slime Mould Algorithm (SMA) and Harris Hawks Optimization (HHO) have shown promising applications in robotics. The Slime Mould Algorithm (SMA) outperformed many algorithms in image registration for mobile robot visual control [10]. HHO effectively optimized 6DOF robot arm kinematics, improving workspace volume and robot conditioning [11]. The latest algorithms, such as the Puma Optimizer (PO) algorithm, have been effectively applied in robotic systems, notably enhancing the control of wheeled mobile robots by improving response time and stability through an enhanced version that incorporates chaotic maps [12], [13].

In this paper, we propose a Puma Optimization-based tuning of PID Control for the Quadruped robot and compare it with two contrast algorithms, SMA and HHO. The PID tuning operation is usually slowed down by many local minima in the n-dimensional solution space, where many combinations of PID gains result in suboptimal solutions. Nevertheless, the PO's adaptive exploration and exploitation strategies navigate the fitness landscape successfully, avoiding local optima and moving towards a globally optimal PID gain setting. Section II of this paper explores the kinematic model; Section III shows the dynamic model; Section IV introduces the Puma optimizer algorithm; Section V outlines the PID controller design and its integration with the algorithms; and Section VI presents numerical results and comparisons. At last, the concluding remarks and future work are given in Section VII.

II. Kinematic Design and Modeling

A quadruped robot with three degrees of freedom guarantees flexibility and stability (Fig 1). With the help of the Multibody Simscape package, accurate movement is made possible by

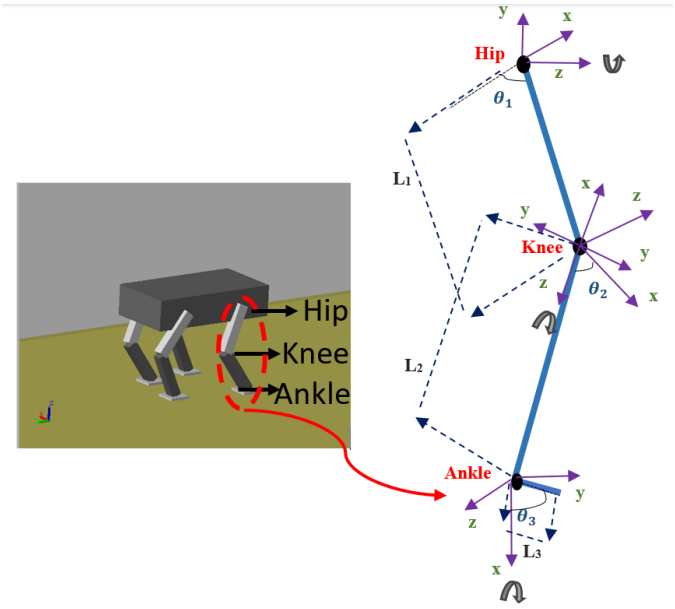


Fig. 1: CAD Model of the quadruped robot (left) and DH coordinate diagram of the leg of a quadruped robot (right)

independently controlled joints. Kinematic and dynamic analysis is centered on the sagittal plane, and a two-beat gait is intended for human-like walking. Because of their function in flexion and extension, the major joints—the knee, hip, and ankle pitches—were chosen. Both the left and the right legs are included in this analysis.

A. Forward Kinematics of the Single Leg

Kinematic analysis is essential for quadruped robot gait planning, and spatial relationships are accurately described using the Denavit–Hartenberg (D-H) approach. Table I lists the D-H parameters of a single leg. Based on the D-H convention, the overall transformation ${}^{n}T_{n+1}$ is obtained through four motion transformation matrices, as given in (1), which form the basis for the forward kinematics.

TABLE I: DH Parameters

#	θ (angle)	d (offset)	a (link)	α (twist)
1	θ_1	0	0	0
2	θ_2	0	L_1	0
3	θ_3	0	L_2	0
4	0	0	L_3	0

$$T = {}^{0}T_1\,{}^{1}T_2\,{}^{2}T_3 = \begin{bmatrix} C_{123} & -S_{123} & 0 & L_3C_{123} + L_2C_{12} + L_1C_1 \\ S_{123} & C_{123} & 0 & L_3S_{123} + L_2S_{12} + L_1S_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

wherein C_{123} equals $\cos(\theta_1+\theta_2+\theta_3)$ and S_{123} is equal to $\sin(\theta_1+\theta_2+\theta_3)$. Utilizing transformation matrices, we can correctly simulate the robot's motion and guarantee control precision when locomotion or maneuvering. This matrix supports the joining of various segments, which then allows coordinated and effective movement patterns.

B. Inverse Kinematics of Single Leg

Taking the right front leg of a quadruped robot as an example, the inverse kinematics equation was deduced. The equation of inverse kinematics of the front leg is as follows:

$$\emptyset = \tan^{-1}\left(\frac{\sin(\theta_{123})}{\cos(\theta_{123})}\right) \quad (2)$$

$$\theta_3 = \emptyset - (\theta_1 + \theta_2) \quad (3)$$

$$\cos(\theta_2) = \frac{\left((x - l_3\cos(\emptyset))^2 + (y - l_3\sin(\emptyset))^2 - l_1^2 - l_2^2\right)}{2l_1l_2} \quad (4)$$

$$\sin(\theta_2) = \pm\sqrt{1 - \cos^2(\theta_2)} \quad (5)$$

$$\theta_2 = \tan^{-1}\left(\frac{\sin(\theta_2)}{\cos(\theta_2)}\right) \quad (6)$$

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{k_2}{k_1}\right) \quad (7)$$

$$x = k_1\cos(\theta_1) - k_2\sin(\theta_1) \quad (8)$$

$$y = k_1\sin(\theta_1) + k_2\cos(\theta_1) \quad (9)$$

where $\emptyset = \theta_1 + \theta_2 + \theta_3$, $k_1 = l_1 + l_2\cos(\theta_2) + l_3\cos(\theta_{23})$, and $k_2 = l_2\sin(\theta_2) + l_3\sin(\theta_{23})$.

III. Dynamic Modeling

The dynamic modeling of quadruped robots involves analyzing the forces and torques acting on the robot's structure, considering both internal and external influences such as inertia, friction, gravity, and joint constraints. Overall dynamics are governed by the interaction of the robot's links, joints, and actuators, ensuring stable locomotion and control in various environments. The Lagrangian approach, comprising of kinetic ($\mathcal{K}_i$) energy and potential energy ($\mathcal{P}_i$) is utilized to model the motion of a quadruped robot as in the following equation:

$$\mathcal{K}_i = \frac{1}{2}m_i\left(\dot{x_l}^2 + \dot{y_l}^2\right) \quad (10)$$

$$\mathcal{P}_i = m_i g y_i \quad (11)$$

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_l} - \frac{\partial L}{\partial \theta_I} \quad (12)$$

$$\tau = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta}) + G(\theta) \quad (13)$$

where m_j = mass of link j (thigh, calf, foot), θ = Vector of joint angles, $\dot{\theta}$ = Vector of joint angular velocities, g = Acceleration due to gravity, $\mathcal{L} = \sum \mathcal{K}_i - \sum \mathcal{P}_i$ for $i = 1$ to n, and n denotes the number of joints, $M(\theta)$ = Inertia matrix, $C\left(\theta,\dot{\theta}\right)$ = Coriolis/centrifugal force matrix , $G(\theta)$ = Gravity force matrix.

The mass matrix $M(\theta)$ defines the inertial properties of the system, capturing how mass is distributed and how it influences motion under applied forces. It is crucial in determining the system's response to external inputs.

$$M(\theta) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (14)$$

$$
\begin{aligned}
m_{11} =& m_1 l_1^2 + m_2\left(l_1^2 + l_2^2 + 2 l_1 l_2 \cos\theta_2\right) + \\
& m_3\left(l_1^2 + l_2^2 + l_3^2 + 2 l_1 l_2 \cos\theta_2 + \right. \\
& \left. 2 l_2 l_3 \cos\theta_3 + 2 l_1 l_3 \cos\left(\theta_2 + \theta_3\right)\right) \\
m_{12} =& m_2\left(l_2^2 + 2 l_1 l_2 \cos\theta_2\right) + m_3\left(l_2^2 + l_3^2 + \right. \\
& \left. l_1 l_2 \cos\theta_2 + l_2 l_3 \cos\theta_3 + l_3 \cos\left(\theta_2 + \theta_3\right)\right) \\
m_{13} =& m_3\left(l_3^2 + l_2 l_3 \cos\theta_3 + l_1 l_3 \cos\left(\theta_2 + \theta_3\right)\right) \\
m_{22} =& m_2 l_2^2 + m_3\left(l_2^2 + l_3^2 + 2 l_2 l_3 \cos\theta_3\right) \\
m_{23} =& m_3\left(l_3^2 + l_2 l_3 \cos\theta_3\right) \\
m_{33} =& m_3 l_3^2
\end{aligned}
$$

and $m_{12} = m_{21}, m_{13} = m_{31}, m_{23} = m_{32}$.

The Coriolis and centrifugal force vector $C(\theta, \theta)$ arise due to velocity-dependent interactions within the system. These forces influence dynamic behavior by introducing additional terms that affect motion stability and control.

$$
C(\theta, \dot{\theta}) = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \tag{15}
$$

$$
\begin{aligned}
c_1 =& -m_2 l_1 l_2 \sin\left(\theta_2 \dot{\theta}_2^2\right) - m_3 l_1 l_2 \sin\left(\theta_2\right) \\
& - m_3 l_1 l_3 \sin\left(\theta_2 + \theta_3\right)\left(\dot{\theta}_2 + \dot{\theta}_3\right) \\
c_2 =& m_2 l_1 l_2 \sin\left(\theta_2 \dot{\theta}_1^2\right) - m_3 l_2 l_2 \sin\left(\theta_3 \dot{\theta}_3^2\right) \\
& - m_3 l_1 l_2 \sin\left(\theta_2 \dot{\theta}_1^2\right) \\
c_3 =& m_3 l_1 l_3 \sin\left(\theta_2 + \theta_3\right) \dot{\theta}_1^2 - l_2 l_3 \sin\left(\theta_3 \dot{\theta}_2^2\right)
\end{aligned}
$$

The gravity vector $G(\theta)$ accounts for the gravitational forces acting on the system, ensuring accurate modeling of the weight-induced effects on each link. This term is essential for compensating gravitational loads during motion.

$$
G(\theta) = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \tag{16}
$$

$$
\begin{aligned}
g_1 =& m_2 l_1 g \cos\left(\theta_1\right) - m_2 g\left(\left(l_1 \cos\theta_1\right)\right. \\
& \left. + l_2 \cos\left(\theta_1 + \theta_2\right)\right) - m_3 g\left(l_3 l_1 \cos\theta_1\right. \\
& \left. + l_2 \cos\left(\theta_1 + \theta_2\right) + l_3 \cos\left(\theta_1 + \theta_2 + \theta_3\right)\right. \\
g_2 =& m_2 l_2 g \cos\left(\theta_1 + \theta_2\right) + m_3 g\left(l_2\left(\cos\left(\theta_1 + \right.\right.\right. \\
& \left.\left.\left. \theta_2\right) l_3 \cos\left(\theta_1 + \theta_2 + \theta_3\right)\right)\right. \\
g_3 =& m_3 g l_3 \cos\left(\theta_1 + \theta_2 + \theta_3\right))
\end{aligned}
$$

IV. Puma Optimizer-based PID Control

This section provides a descriptive analysis of the PO methodology. The algorithm proposed is employed to adjust the gains of the PID controller, enabling precise trajectory control along a specified path for a quadruped robot. The Puma Optimizer, a hyper-heuristic algorithm, is inspired by the hunting behavior of the pumas and employs a novel approach to alter the phases of exploration and exploitation. In the PO algorithm, the optimal solution is considered a male puma, while the entire optimization domain is viewed as the puma's territory. Also, other solutions (Xi) have been considered as the female puma.

This algorithm works in two stages, that is, the *inexperienced* and *experienced stages*. In the *inexperienced stage* of the Puma algorithm, consisting of three iterations, the exploration and exploitation operations are performed simultaneously until the initialization is done [13]. The exploration phase in Puma optimization broadens the solution space to find diverse candidate solutions. In contrast, the exploitation phase refines the best solutions by intensively searching around them to achieve optimality. In the *experienced stage*, two different functions, $g1$ and $g2$, are used for scoring. These functions are the evaluation metrics that help prioritize between exploration and exploitation based on cost and time-related sequences. $g1$ highlights the escalation aspect, prioritizing the phase, exploration or exploitation, that has demonstrated superior performance, while $g2$ emphasizes the resonance component and leads to the phase that performs better than the other prioritized phase.

$$
g1_{\text{explore}} = PG_1\left(\frac{Sq^1_{\text{CExplore}}}{Sq_t}\right) \tag{17}
$$

$$
g1_{\text{exploit}} = PG_1\left(\frac{Sq^1_{\text{CExploit}}}{Sq_t}\right) \tag{18}
$$

$$
g2_{\text{explore}} = PG_2\left(\frac{Sq^1_{\text{CExplore}} + Sq^2_{\text{CExplore}} + Sq^3_{\text{CExplore}}}{Sq^1_t + Sq^2_t + Sq^3_t}\right) \tag{19}
$$

$$
g2_{\text{exploit}} = PG_2\left(\frac{Sq^1_{\text{CExploit}} + Sq^2_{\text{CExploit}} + Sq^3_{\text{CExploit}}}{Sq^1_t + Sq^2_t + Sq^3_t}\right) \tag{20}
$$

$\text{Sq}_{\text{CExplore}}$ and $\text{Sq}_{\text{CExploit}}$ are the sequence cost variables which represents the weighted sum of exploration and exploitation respectively to decide between exploration and exploitation phase. Sq_t is also a variable with a constant value. PG_1 and PG_2 are parameters with a fixed value equal to 0.5, which is initialized before the optimization procedure and are used to prioritize each of the functions $g1$ and $g2$. If score of exploitation is greater than that of exploration, then it enters the exploitation phase and otherwise enters the exploration phase.

A. Exploration

In the exploration phase, the behavior of pumas in their search for food serves as inspiration. At first, the entire population is sorted in ascending order, then Puma improves its solutions in the exploration phase.

$$
Y_{\text{New}} = \begin{cases} W_{m,H} & \text{if } k = k_{\text{random}} \text{ or } \text{random}_1 \leq V \\ B_{c,H} & \text{otherwise} \end{cases} \tag{21}
$$

$$NC' = 1 - V \tag{22}$$

$$q = \frac{NC'}{N_{\text{group}}} \tag{23}$$

$$\text{if } \text{Cost}\left(Y_{\text{new}}\right) < \text{Cost}\left(Y_m\right),$$

$$V = V + q \tag{24}$$

where k_{random} is a randomly generated integer within the dimensional scope of the problem and $random_1$ is also a randomly produced number in uniform distribution generated between 0 and 1. V is a parameter value in the range $[0, 1]$, and Y_{new} represents the updated solution based on dimensional randomness. N_{group} denotes the total number of pumas, and N'_C represents the number of unchanged dimensions. $W_{m,H}$ is a randomly generated solution used for exploration within the search space, while $B_{c,H}$ indicates the position of the solution c in the population at generation H. The parameter q denotes the proportion of unchanged dimensions.

This action avoids the local optimum and the product solutions are diverse. The newly generated solutions are replaced with the current solution.

$$B_{c,H} = Y_{\text{New}}, \text{ if } Y_{m,\text{New}} < B_{c,H} \tag{25}$$

where $Y_{m,New}$ represents the newly generated solution that is evaluated for replacement in the optimization process.

B. Exploitation

In the exploitation phase, the PO algorithm utilizes two different operators to improve the solutions, and these two mechanisms are based on the two behaviours of pumas: hunting using ambush and sprinting.

$$Y_{\text{New}} = \begin{cases} \dfrac{\text{mean}(\text{Sol}_{\text{sum}})}{N_{\text{group}}\left(Y_p^l - (-1)^{\gamma} \cdot Y_m\right)}, \\ \text{if random}_2 > 0.5 \\ \\ \text{Puma}_{\alpha} + (2 \cdot \text{random}_3) \cdot \exp(\text{random}_{n1}) \cdot Y_q^l - Y_m, \\ \text{if random}_4 \geq 1 \\ \\ (2 \cdot \text{random}_5) \cdot \frac{(\text{G}_1 \cdot S \cdot Y(m) + G_2 \cdot (1-S) \cdot \text{Puma}_{\alpha})}{2 \cdot \text{random}_6 - 1 + \text{random}_{n2}} \cdot \text{Puma}_{\alpha}, \\ \text{otherwise} \end{cases} \tag{26}$$

Where Sol_{sum} refers to the sum of all solutions. Y_p^l is the random solution in the whole population. B value is equal to 0 or 1 produced randomly by s. Y_m is the current solution, and Puma_{α} refers to the best solution for the entire population. S is a parameter that is used to control the influence of a random search component, and the parameters G_1 and G_2 are used to control the influence of the best solution. $random_2$ to $random_6$ are random values between 0 and 1 that are essential to maintain diversity and avoid premature convergence. $random_{\text{n1}}$ and $random_{\text{n2}}$ are randomly generated numbers in the normal distribution and the dimensions of the problem. PO has a regularly low computational complexity and equals $O\left((2T+1)N^2 * N * D\right)$ where N is the population size (number of Pumas), T is the total number of iterations, and D is the dimensionality of the problem (number of decision variables).

V. Optimal PID Control

The control architecture in Fig. 2 consists of an offline tuning phase where simulations are run for various PID parameter sets (K_p, K_i, K_d) generated by an optimization algorithm. The initial optimization parameters, including iterations, agent count, PID bounds, and cost weights, are listed in Table II. The PID controller output is a sum of three terms: proportional, integral, and derivative. Each of these terms depends on the error value e between the input and the output.

$$\text{Output } = K_p \cdot e(t) + K_i \cdot \int_0^t e(t)dt + K_d \cdot \frac{de}{dt} \tag{27}$$

The joint angular error and output torque from each simulation are used as feedback to compute the cost, guiding the optimization process toward optimal PID values. Once the best parameters are identified, they are fed into the PID controller to evaluate the system's response. This process is repeated for all three optimization algorithms to compare their effectiveness.

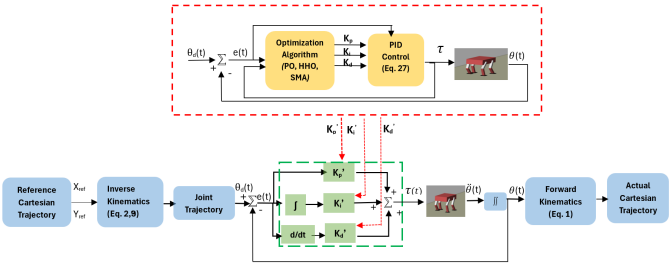


Fig. 2: Control system architecture for the quadruped robot where K'_p, K'_i, and K'_d are the optimal PID gain values.

TABLE II: Initial Parameters

Parameters	Values
Iterations	40
Agents	25
Upper Bounds (K_p, K_i, K_d)	(15000, 5000, 250)
Lower Bounds (K_p, K_i, K_d)	(1000, 500, 10)
Cost Function Weights $(\alpha,\ \beta,\ \gamma,\ \delta)$	(0.4, 0.3, 0.2, 0.1)

The performance of the PID controller is evaluated using a weighted cost function:

$$J = \alpha \int_0^t t|e(t)|dt + \beta \int_0^t |e(t)|dt + \delta \int_0^t e^2(t)dt + \gamma \int_0^t |\tau(t)dt| \tag{28}$$

where $e(t)$ represents the joint angular error and τ denotes the applied torque. The cost function comprises of four key terms: the Integral of Time-weighted Absolute Error (ITAE), which penalizes errors persisting over time; the Integral of Absolute Error (IAE), which minimizes overall deviation; the Integral of Squared Error (IASE), which suppresses large spikes; and the Integral of Absolute Torque (IAT), which discourages excessive control effort.

VI. Results and Discussions

The PO algorithm, along with two other algorithms (HHO and SMA), are programmed in the MATLAB environment to find the optimal values of tuning parameters [10], [11]. The proposed algorithms run offline for 1000 iterations to find the minimum value of the cost function (J) and tuning time (T). After tuning, the best set of PID gains corresponding to the minimum cost values for all three algorithms is given in Table III. Graphical results from these simulations are obtained.

TABLE III: Optimal PID Gains for all three Controllers

HHO				
K_p	K_i	K_d	$Cost(J)$	$T(min)$
9747.44	4444.56	138.27	0.2443	128.4
SMA				
K_p	K_i	K_d	$Cost(J)$	$T(min)$
7481.79	1702.72	49.60	0.0712	112.5
PO				
K_p	K_i	K_d	$Cost(J)$	$T(min)$
10107.24	2583.33	91.66	0.0518	174.2

The Cartesian trajectory performances of the algorithms are traced in X and Y directions and are shown in Fig 3(a) and 3(b), respectively. The controllers tuned with the optimization algorithms ensure the desired path is traced successfully while providing similar responses. However, in Fig 4(a) and 4(b), the PO algorithm shows fewer errors of 0.0335m and 0.0463m in X and Y direction respectively, demonstrating precise and stable control with less deviation from the desired path.

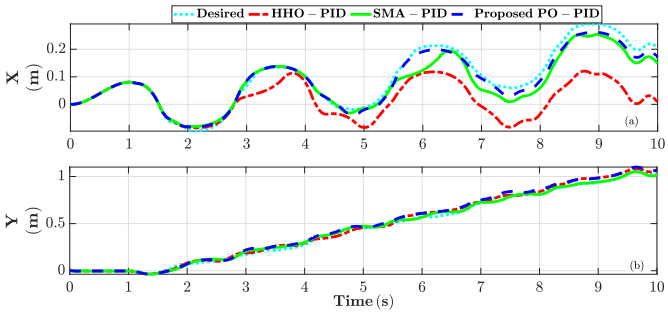


Fig. 3: Cartesian trajectory of the quadruped robot leg (a) X-direction and (b) Y-direction.

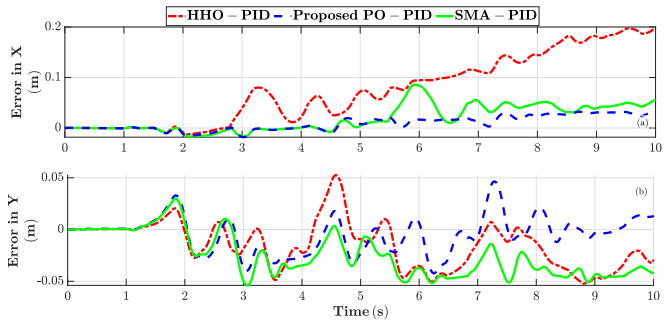


Fig. 4: Tracking error of quadruped leg (a) X-direction and (b) Y-direction.

Figs. 5(a)-5(c) shows the angular position of all three joints—hip, knee, and ankle—run under three optimization algorithms compared with the desired trajectory. The three algorithms performed well in tracing the desired joint trajectory. Fig. 6(a)-6(c) highlight the PO algorithm's exceptional performance, showing the least errors compared to the other two algorithms. Specifically, the maximum absolute errors in joint trajectories were 0.00313 for HHO, 0.00273 for SMA, and 0.00205 for the PO algorithm.

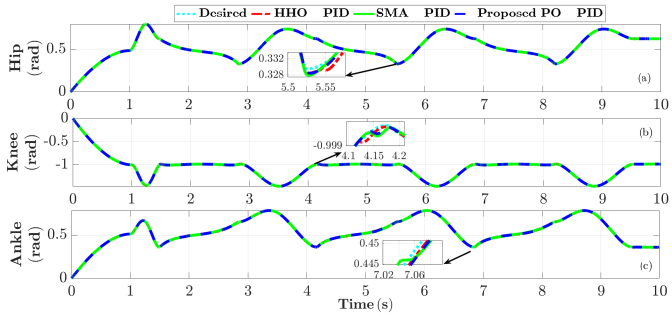


Fig. 5: Angular position of the quadruped leg (a) hip, (b) knee, and (c) ankle.

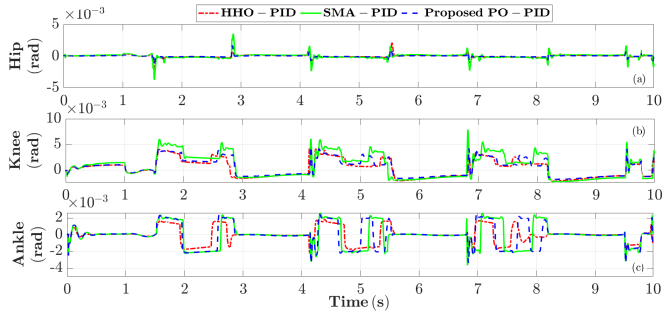


Fig. 6: Angular error of the quadruped leg (a) hip, (b) knee, and (c) ankle.

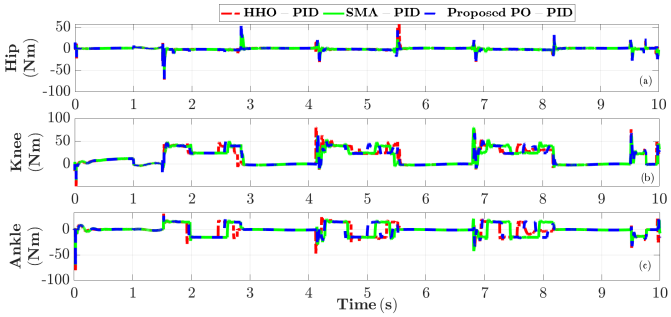


Fig. 7: Joint torque of the quadruped leg (a) hip, (b) knee, and (c) ankle.

Figs. 7(a)- 7(c) illustrate the control efforts of all the joints while comparing the three algorithms in producing the output control signal. It can be concluded that the SMA and PO-based PID controllers require a similar amount of torque, whereas

the HHO-based controller requires slightly more torque. The PO-based PID controller provided a smoother output, thereby ensuring better stability during tuning. Specifically, the maximum torque values observed for the PO-based PID controller were 53.1069 Nm at the hip, 68.0304 Nm at the knee, and 27.4609 Nm at the ankle joint.

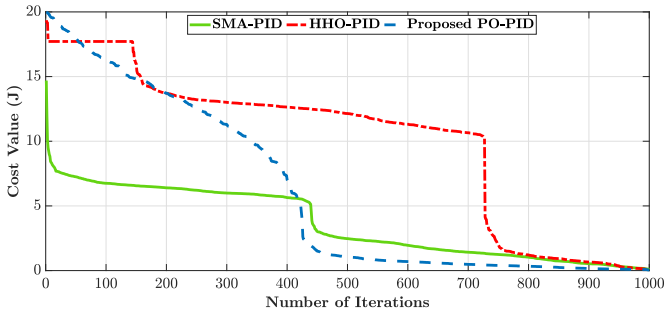
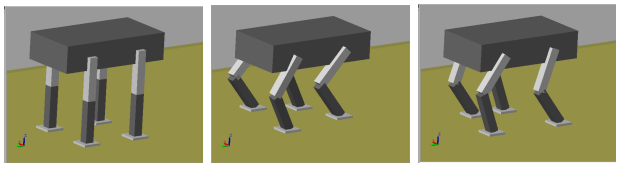


Fig. 8: Convergence curve of best cost values for the three algorithms

The objective function convergence performances of all three algorithms are graphically illustrated in Fig 8. It can be seen that the PO algorithm converged better than the other two. In other words, the Puma Optimization algorithm proved to be better at finding the best solutions and reaching the global minimum. Table IV illustrates the performance metrics of the three algorithms. It can be seen that the PO-PID control resulted in the lowest joint and Cartesian trajectory errors among the three algorithms, highlighting its effectiveness.

TABLE IV: Performance Metrics for Optimal-PID Controllers

Type of Error	**Contrast and Proposed Control**		
	HHO	**SMA**	**PO**
Joint Trajectory Errors (rad.)			
RMS	0.0012	0.0006	0.0004
MAE	0.0031	0.0027	0.0020
ITAE	0.0313	0.0273	0.0205
Cartesian Trajectory Errors (m)			
RMS(X)	0.0823	0.0263	0.0135
RMS(Y)	0.0199	0.0254	0.0137
MAE(X)	0.1969	0.0858	0.0335
MAE(Y)	0.0525	0.0539	0.0463
ITAE(X)	3.7285	1.1035	0.5924
ITAE(Y)	0.7928	1.1147	0.4326



1 sec 5 sec 10 sec

Fig. 9: Simulation of the quadruped robot with PO-PID control

Fig. 9 shows a simulation of the quadruped's locomotion at 1, 5, and 10 seconds under PO-PID control. The robot begins from rest and gradually initiates smooth movement. Specifically, these simulation snippets highlight the controller's capability to manage dynamic transitions while maintaining system stability and delivering precise actuation of the robot's legs throughout the movement cycle.

VII. Conclusions

This paper focused on optimizing PID controllers for a quadruped robot to achieve precise footstep control along a desired trajectory. The PO algorithm was introduced for the first time as a diversity-based tuning method, and its performance was compared with the widely used SMA and HHO algorithms. Results showed that PO outperformed SMA and HHO in global and local searches, making it the most effective optimization method among the three. However, all tuned PID controllers successfully controlled the robot's trajectory with comparable performance. This study contributed to advancing quadruped robot walking and control strategies. Future work could explore hybrid optimization approaches (combining multiple algorithms) to further enhance PID tuning for improved quadruped robot stability and adaptability.

References

[1] M. Li, Z. Liu, M. Wang, G. Pang, and H. Zhang, "Design of a parallel quadruped robot based on a novel intelligent control system," *Applied Sciences*, vol. 12, no. 9, p. 4358, 2022.

[2] Y. Fan, Z. Pei, C. Wang, M. Li, Z. Tang, and Q. Liu, "A review of quadruped robots: structure, control, and autonomous motion," *Advanced Intelligent Systems*, vol. 6, no. 6, p. 2300783, 2024.

[3] B. Chol and S. Sreenivasan, "Motion planning of a wheeled mobile robot with slip-free motion capability on a smooth uneven surface," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 4. IEEE, 1998, pp. 3727–3732.

[4] W. Zhang and K. Wang, "Wheely: An accessible platform for experimenting with wheeled-legged quadrupedal robots," in *2023 3rd International Conference on Computer, Control and Robotics (ICCCR)*. IEEE, 2023, pp. 220–224.

[5] D. Wang, W. Cao, H. Yu, I. Taima, and A. Takanishi, "Enhancing elderly mobility with a quadruped robot-led wheelchair system based on mpc," in *2024 14th Asian Control Conference (ASCC)*. IEEE, 2024, pp. 1–6.

[6] C. Choubey and J. Ohri, "Gwo-based tuning of lqr-pid controller for a 3-dof parallel manipulator," *IETE Journal of Research*, vol. 69, no. 7, pp. 4378–4393, 2023.

[7] D. Ma, S.-I. Niculescu, L. Guo, and J. Chen, "Special issue on pid control in the information age: Theoretical advances and applications." *International Journal of Robust & Nonlinear Control*, vol. 32, no. 18, 2022.

[8] M. Latifinavid and A. Azizi, "Kinematic modelling and position control of a 3-dof parallel stabilizing robot manipulator," *Journal of Intelligent & Robotic Systems*, vol. 107, no. 2, p. 17, 2023.

[9] S. A. Rashad, M. Sallam, A. Bassiuny, and A. Abdel Ghany, "Control of master slave robotics system using optimal control schemes," in *International Conference on Aerospace Sciences and Aviation Technology*, vol. 18, no. 18. The Military Technical College, 2019, pp. 1–12.

[10] D. İzci and S. Ekinci, "Comparative performance analysis of slime mould algorithm for efficient design of proportional-integral-derivative controller." *Electrica*, vol. 21, no. 1, 2021.

[11] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future generation computer systems*, vol. 97, pp. 849–872, 2019.

[12] M. Kmich, N. El Ghouate, A. Bencharqui, H. Karmouni, M. Sayyouri, S. Askar, and M. Abouhawwash, "Chaotic puma optimizer algorithm for controlling wheeled mobile robots," *Engineering Science and Technology, an International Journal*, vol. 63, p. 101982, 2025.

[13] B. Abdollahzadeh, N. Khodadadi, S. Barshandeh, P. Trojovskỳ, F. S. Gharehchopogh, E.-S. M. El-kenawy, L. Abualigah, and S. Mirjalili, "Puma optimizer (po): a novel metaheuristic optimization algorithm and its application in machine learning," *Cluster Computing*, vol. 27, no. 4, pp. 5235–5283, 2024.