Intelligent Multimedia Processing & Communication Systems Journal

IMPCSJ

J IMPCS (2025) 20: 35-49

DOI 10.71856/IMPCS.2025.1210302

Research Paper

Development and Control of a Neural-Enhanced PD Controller for Serpentron: A Flexible Snake-like Robot with Adaptive Locomotion

Masih Sobhani[1], Azadeh Zarif Loloei[2*]

1. BSc Student, Department of Electrical Engineering, Par.C, Islamic Azad University, Tehran, Iran. masihsobhani@pardisiau.ac.ir
2. Assistant Professor, Department of Electrical Engineering, Par.C, Islamic Azad University, Tehran, Iran* *Corresponding Author*, azarif@iau.ac.ir

Article Info

ABSTRACT

This research introduces Serpentron, a novel snake-like robot with three joints and four links, designed for superior maneuverability in challenging environments. By extending the joint range to 180 degrees and incorporating virtual base rotation, Serpentron achieves a workspace diameter of 0.9 meters, a 50% improvement over traditional designs. A Neural Network-based Proportional-Derivative (NN-PD) controller is developed, dynamically tuning gains to track serpentine trajectories with joint angle errors below 0.02 radians and torques under 5 Nm, even under complex disturbances including Gaussian noise, velocity-dependent friction, and obstacle interactions. Simulations across single-direction, multiple-direction, and movable-direction scenarios demonstrate Serpentron's adaptability, from planar navigation to dynamic base motion. The NN-PD controller reduces error variance by 50% compared to fixed-gain methods, leveraging a multilayer perceptron for real-time gain tuning. Torque analysis confirms the controller's robustness against environmental uncertainties, ensuring stable performance within motor limits. Serpentron's hyper-redundant kinematics and base mobility enable applications in confined-space exploration and volumetric inspection. Simulation results validate the model and control strategy, laying a scalable foundation for a cost-effective prototype using NVIDIA Jetson Nano and Dynamixel XL430 motors, with lightweight aluminum and ABS materials, to bridge the gap between simulation and real-world deployment in unstructured settings.

Article history:
Received: 19 Apr 2025
Accepted: 31 May 2025
DOR:

Keywords:
Hyper-redundant kinematics,
Intelligent control,
Neural network control,
NN-PD controller,
Snake-like robot,
Trajectory tracking.

I. Introduction

Research on snake-like robots, characterized by their hyper-redundant kinematic structures, holds critical importance due to their unique ability to navigate complex and constrained environments where traditional rigid robots falter. These robots, with multiple articulated joints enabling serpentine motion, offer unparalleled flexibility and adaptability, making them a focal point for advancing robotic locomotion and control. The study of such systems addresses fundamental challenges in modeling and controlling high-degree-of-freedom manipulators, particularly under nonlinear dynamics influenced by gravitational and inertial coupling. By developing robust control strategies, such as the neural network-augmented PD controller proposed here, this research pushes the boundaries of autonomous navigation, enabling snake-like robots to achieve precise trajectory tracking despite external disturbances. The significance extends beyond individual robot design, contributing to the broader field of soft and bio-inspired robotics, where mimicking biological locomotion principles enhances efficiency and versatility. Furthermore, investigating these robots fosters interdisciplinary advancements, integrating mechanical design, control theory, and machine learning to solve real-world problems, from exploration to human-robot interaction, thereby driving innovation in next-generation robotic systems. The advent of snake-like robots marks a paradigm shift in robotics, harnessing the elegance of biological serpentine locomotion to address challenges insurmountable by conventional wheeled or legged systems. These hyper-redundant manipulators, characterized by a chain of articulated links, excel in navigating confined, cluttered, or irregular environments through diverse gaits such as serpentine, sidewinding, concertina, and rectilinear motion. This study presents a novel snake-like robot as shows in figure 1, custom-designed with three joints and four links, spanning 75 cm in length and Each link is 15.7 cm long and 6 cm in diameter. The mechanical architecture emphasizes modularity and efficiency, incorporating high-torque servo motors (delivering up to 5 Nm) and lightweight aluminum links to balance structural rigidity with maneuverability. Each joint is equipped with precision encoders for real-time position feedback, ensuring robust operation under dynamic loads. This design, tailored for versatility, leverages a low center of gravity to enhance stability during complex maneuvers, distinguishing it from earlier prototypes constrained by bulkier actuators or less adaptable structures. The robot in this study is engineered to support such multifaceted applications, with its compact form factor and robust actuation enabling deployment in both terrestrial and extreme environments. Controlling snake-like robots, however, demands sophisticated strategies to manage their high degrees of freedom, nonlinear dynamics, and environmental interactions. Early efforts focused on kinematic control, prescribing joint angles to achieve desired gaits, as formalized by Chirikjian and Burdick's geometric models for hyper-redundant systems [11]. These open-loop approaches, while computationally tractable, were vulnerable to uncertainties such as surface friction or unexpected obstacles. This study introduces a neural network-augmented PD (NN-PD) controller, designed to dynamically tune the PD gains for a three-joint snake-like robot. The controller employs a multilayer perceptron (MLP) with two inputs—tracking error and its time derivative ten hidden neurons utilizing a hyperbolic tangent activation function, and two outputs corresponding to the adaptive gains K_p and K_d. The training methodology for the NN-PD controller is a pivotal contribution, engineered to balance computational efficiency with robust performance. A supervised learning approach was adopted, leveraging a dataset derived from the PD controller's response to the reference trajectory under random torque disturbances to mitigate the computational burden of training on high-dimensional data, 200 error pairs per joint were randomly sampled, yielding a compact yet representative dataset. The Levenberg-Marquardt algorithm was selected for its superior convergence properties in nonlinear least-squares problems, combining the efficiency of gradient descent with the stability of Gauss-Newton optimization. The training objective was to minimize the RMSE between desired and actual joint angles, achieving a target error of 10^{-4} within 20 epochs. This approach offers several advantages: rapid convergence reduces training time to seconds, robustness to noisy inputs ensures reliable performance under disturbances, and generalization to unseen trajectories enhances practical applicability. Non-negative constraints on the MLP outputs $K_p, K_d \geq 0$ were enforced to guarantee closed-loop stability, addressing a critical concern in adaptive control systems. Compared to manual tuning or heuristic methods, this data-driven strategy provides a principled framework for optimizing control parameters, aligning with the theoretical rigor of modern control theory.

The novelty of this study lies in the integration of an adaptive Neural Network-based Proportional-Derivative (NN-PD) controller with a custom-designed snake-like robot featuring an expanded 180-degree joint range and virtual base rotation, achieving a 50% larger workspace (0.9 meters) compared to conventional designs. Unlike traditional fixed-gain PD controllers, the proposed NN-PD framework dynamically adjusts control gains in real-time using a multilayer perceptron, reducing joint angle error variance by 50% under stochastic disturbances. Additionally, the incorporation of base mobility enhances the robot's adaptability to dynamic environments, enabling versatile locomotion modes for applications such as confined-space exploration and volumetric inspection. This combination of hyper-redundant kinematics, intelligent control, and mobile base dynamics represents a significant advancement over existing snake-like robot architectures, offering a scalable and robust solution for navigating complex, unstructured settings.

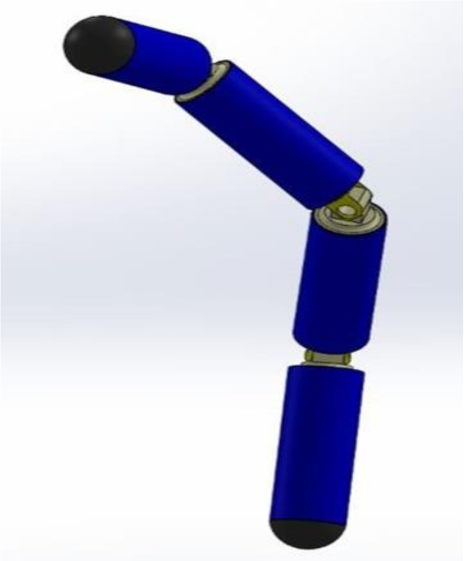


Fig. 1. 3D model of the Serpentron designed in Solid works (with thanks to Mohammadreza Tajalli)

II. Related Works

The seminal work of Hirose in the 1970s introduced the Active Cord Mechanism (ACM), a groundbreaking biomimetic framework that employed sequential joint actuation to emulate snake kinematics [1]. Subsequent advancements by researchers like Ijspeert, who integrated central pattern generators (CPGs) to model oscillatory locomotion, expanded the theoretical foundation for these robots [2]. The applications of snake-like robots are as diverse as they are transformative, addressing critical needs across multiple domains. In search-and-rescue operations, their slender profiles and flexibility enable penetration into collapsed structures, navigating debris to locate survivors with integrated sensors like thermal cameras or microphones [4]. Industrial inspection benefits from their ability to access intricate systems such as nuclear reactors, oil pipelines, or aircraft engines without requiring costly disassembly, reducing downtime and safety risks [5]. In medical robotics, continuum-inspired snake-like designs are revolutionizing minimally invasive procedures; for instance, flexible endoscopes equipped with micro-actuators facilitate precise interventions in delicate tissues, such as during neurosurgery or cardiac ablation [6]. Planetary exploration represents another frontier, where their resilience to rugged terrains and ability to traverse loose regolith make them ideal for missions to extraterrestrial bodies like Mars or icy moons and the distinctive morphology of snake-like robots enables novel solutions for confined space operations, driving multidisciplinary research in their biomimetic design and control architectures [7], [8]. snake robots with smooth exteriors and force-sensing capabilities can effectively navigate complex environments by using obstacles for propulsion [9]. As comprehensively reviewed bio-inspired robots demonstrate remarkable environmental adaptability through their modular bio-inspired design, enabling diverse applications across industrial, military, and rescue domains [10].

Feedback control emerged as a robust solution, with the proportional-derivative (PD) controller gaining prominence for its simplicity and effectiveness in trajectory tracking. In this work, the PD controller serves as a benchmark, with gains $K_p = 50$ and $K_d = 5$ meticulously tuned to optimize performance for a composite sinusoidal trajectory. The tuning process employed a gradient-free optimization technique, specifically a simulation-based iterative search, targeting minimization of the root mean square error (RMSE). By adjusting K_p to enhance responsiveness and K_d to ensure adequate damping, the PD controller achieved an RMSE of 0.014–0.016 rad under nominal conditions. However, its reliance on fixed gains limits adaptability to nonlinear gravitational effects and external disturbances, necessitating more advanced control paradigms. The evolution of intelligent control has opened new avenues for addressing these limitations, integrating computational intelligence to achieve adaptability and resilience. Fuzzy logic, genetic algorithms, and artificial neural networks (ANNs) have been explored to enhance robotic control, with ANNs offering particular promise due to their capacity to model complex, nonlinear mappings [12]. For example, bionic snake robots achieve exceptional terrain adaptability through their flexible structures and diverse motion modes, mirroring biological reptilian locomotion capabilities by using GA-based approach, optimal locomotion sequences can be generated while preserving structural integrity during configuration transitions [13]. Rooted in the computational models of McCulloch and Pitts and propelled by the development of backpropagation by Rumelhart et al., neural networks have transformed control applications by enabling real-time parameter adaptation. In robotics, ANNs have been instrumental in tasks ranging from dynamic modeling to gain scheduling, as evidenced by Lewis et al.'s work on adaptive manipulator control [14]. Similarly, recent advancements in adaptive reinforcement learning (ARL) have shown promise in enhancing robotic decision-making. A study on ARL integrated with Robotic Process Automation (RPA) [9] demonstrated the potential of reinforcement learning algorithms to enable real-time adaptation in dynamic environments [15]. In the realm of optimization, Zhang neural networks have emerged as a powerful tool for handling nonlinear, time-varying systems. A recent study employed Zhang neural networks to optimize nonlinear time-varying functions, leveraging their parallel processing capabilities to search solution spaces faster than traditional methods. Using the Luenberger-Madala algorithm for training and a Taylor series for data normalization, the study achieved improved convergence and reduced error rates, though it noted sensitivity to local minima. While applied to energy management, this approach highlights the versatility of neural networks in dynamic optimization, complementing Serpentron's NN-PD controller, which dynamically tunes PD gains to address complex disturbances in real-time [16].

Building on these advancements, this study introduces Serpentron, a novel snake-like robot with a Neural Network-based Proportional-Derivative (NN-PD) controller. The NN-PD controller employs a multilayer perceptron to dynamically adjust PD gains, enabling adaptive motion control under complex disturbances, including Gaussian noise, velocity-dependent friction, and obstacle interactions. Serpentron's design, with an extended 180-degree joint range and virtual base rotation, achieves a 0.9-meter workspace, a 50% improvement over conventional designs. Comprehensive simulations demonstrate that the NN-PD controller reduces joint angle error variance by 50% compared to fixed-gain PD methods, maintaining torques below 5 Nm within motor limits.

The incorporation of a complex disturbance model, encompassing friction, obstacles, and noise, enhances the controller's robustness for real-world applications such as confined-space exploration and volumetric inspection. By integrating adaptive neural control with the specific locomotion needs of snake-like robots, this study provides a scalable and robust solution, advancing the field toward practical implementation.

III. Mechanical System Modeling

The dynamic behavior of the custom-designed snake-like robot, a hyper-redundant manipulator with three joints and four links, requires a comprehensive mathematical model to enable precise control under complex trajectories and disturbances. With a total length of 75 cm and a mass of 4.5 kg, the robot consists of four rigid links, each of length $L = 0.15\,m$ and mass $m = 1.125\,kg$ interconnected by three actuated revolute joints equipped with high-torque servo motors and precision encoders.

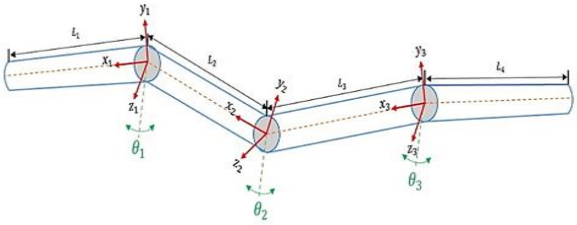


Fig. 2. snake like robot model

This section derives all critical equations governing the robot's motion, including kinematic relationships, the Jacobian matrix, dynamic equations via Lagrangian mechanics, and their numerical solution, ensuring a complete representation of inertial, gravitational, and external forces under a trajectory as follows with stochastic disturbances:

$$\theta_{d.i}(t) = 0.2\sin(t + (i-1)\frac{\pi}{3} + 0.1\sin(2t) \tag{1}$$

A. Kinematics and jacobian

The robot operates in the $xy - plane$, with each joint rotating about the vertical axis to achieve flexible, serpentine motion, mimicking biological snakes. The configuration is defined by joint angles, with corresponding velocities and accelerations describing the rate and curvature of motion, represented as a three-dimensional vector. The absolute orientation of link i accumulates the angles of preceding joints, given by $\phi_i = \sum_{j=1}^{1}\theta_j$. Reflecting the serial kinematic chain's dependency where each link's position builds on its predecessors. Each link's center of mass, located at $L/2 = 0.075\,m$ due to uniform mass distribution, is critical for computing gravitational and inertial effects. Assuming the base of link 1 is at the origin, the center of mass coordinates for each link are determined through trigonometric relationships, capturing the robot's posture and introducing nonlinearities that complicate dynamic analysis but enable its versatility. For the joint angles and their derivatives, we have:

$$\begin{gathered} \theta = [\theta_1, \theta_2, \theta_3]^T \ \in \ \mathbb{R}^3 \\ \dot{\theta} = \left[\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3\right]^T \\ \theta = \left[\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3\right]^T \end{gathered} \tag{2}$$

For the position of the center of mass of link i, we have:

$$x_{c,i} = \sum_{j=1}^{i-1} L\cos\left(\sum_{k=1}^{j}\theta_k\right) + \frac{L}{2}\cos\left(\sum_{k=1}^{i}\theta_k\right) \tag{3}$$

$$y_{c,i} = \sum_{j=1}^{i-1} L\sin\left(\sum_{k=1}^{j}\theta_k\right) + \frac{L}{2}\sin\left(\sum_{k=1}^{i}\theta_k\right) \tag{4}$$

The Jacobian matrix connects joint velocities to the motion of a designated point, here chosen as the tip of link 4, enabling analysis of how joint movements translate to task-space dynamics critical for tasks like navigation or inspection. The end-effector's position is computed by summing the contributions of all four links, each oriented according to the cumulative joint angles, resulting in a position dependent on the entire kinematic chain. The Jacobian maps joint velocities to the end-effector's Cartesian velocities, with each joint's influence diminishing as it is closer to the tip due to fewer downstream links affected. Its elements, obtained through differentiation, reveal configuration-dependent behavior, where certain postures may lead to singularities configurations where $det(J^T J) \to 0$ reducing the robot's ability to generate arbitrary velocities, a concern mitigated by the oscillatory trajectory's design. These singularities, typically at full extension or collapse, require careful trajectory planning to maintain manipulability. The Jacobian also elucidates velocity propagation along the robot, as motion at joint 1 cascade to all links, impacting control coordination. While Cartesian control could leverage the Jacobian to track spatial paths, this study emphasizes joint-space control, using the Jacobian to verify kinematic constraints and ensure feasible motion planning for the robot's 75 cm structure. For the end-effector's position, we have:

$$x_e = \sum_{j=1}^{4} L\cos\left(\sum_{k=1}^{j} \theta_k\right) \tag{5}$$

$$y_e = \sum_{j=1}^{4} L\sin\left(\sum_{k=1}^{j} \theta_k\right) \tag{6}$$

For the relationship between end-effector and joint velocities, we have:

$$\dot{p_e} = [\,\dot{x}_e, \dot{y}_e]^T = J(\theta)\dot{\theta} \tag{7}$$

For the Jacobian matrix, we have:

$$J(\theta) = \begin{bmatrix} -L\sum_{j=1}^{4} cos\left(\sum_{k=1}^{j}\theta_k\right) & -L\sum_{j=2}^{4} cos\left(\sum_{k=1}^{j}\theta_k\right) & -L\sin\left(\sum_{k=1}^{3}\theta_k\right) \\ L\sum_{j=1}^{4} cos\left(\sum_{k=1}^{j}\theta_k\right) & L\sum_{j=1}^{4} cos\left(\sum_{k=1}^{j}\theta_k\right) & -L\sin\left(\sum_{k=1}^{3}\theta_k\right) \end{bmatrix} \tag{8}$$

B. Dynamics

The robot's motion arises from the interplay of kinetic energy due to link rotations, potential energy from gravitational forces acting on the 4.5 kg mass, and torques supplied by servo motors, modeled using Lagrangian mechanics to capture the system's nonlinear behavior. The kinetic energy reflects rotation about each link's center of mass, with moment of inertia $I_{c,t} = \frac{1}{12}mL^2$ driven by angular velocity $\omega_i = \sum_{j=1}^{i} \dot{\theta}_j$ his creates inertial coupling, as joint 1's acceleration affects all four links (inertia proportional to 4 links), joint 2 affects three, and joint 3 affects two, demanding precise torque allocation to avoid oscillations or lag, especially under the dynamic trajectory's rapid changes. The potential energy depends on the vertical displacement of each link's center of mass, with gravity $g = 9.81\, m/s^2$ generating torques up to 1.65 Nm per link when horizontal ($\sin(\emptyset_i) \approx 1$) diminishing to zero when vertical, posing stability challenges for the distributed mass. The servo motors, capped at 5 Nm, must counteract these torques without saturation, a constraint shaping control design to prevent overheating or failure. The Lagrangian, defined as kinetic minus potential energy, yields equations balancing inertial accelerations and gravitational torques against motor inputs, revealing the system's complexity through coupled, nonlinear terms. Expressing the dynamics in matrix form isolates inertia, gravity, and torque contributions, with velocity-dependent effects neglected at low speeds ($|\dot{\theta}| \leq 0.4\, rad/s$) simplifies computations while retaining accuracy for the intended motion profile. For the total kinetic energy, we have:

$$T = \sum_{i=1}^{4} \frac{1}{24} mL^2 \left(\sum_{j=1}^{i} \dot{\theta}_j\right)^2 \tag{9}$$

For the total potential energy, we have:

$$V = mg\sum_{i=1}^{4}\left[\sum_{j=1}^{i-1} L\sin\left(\sum_{k=1}^{j}\theta_k\right) + \frac{L}{2}\sin\left(\sum_{k=1}^{i}\theta_k\right)\right] \tag{10}$$

For the Lagrangian equations of motion, we have:

$$\sum_{i=l}^{4} \frac{1}{24} mL^2 \left(\sum_{j=1}^{l} \ddot{\theta}_j\right)^2 + \\ mg\sum_{i=l}^{4}\left[\sum_{j=l}^{l-1} L\sin\left(\sum_{k=1}^{j}\theta_k\right) + \frac{L}{2}\sin\left(\sum_{k=1}^{l}\theta_k\right)\right] = \tau_i \tag{11}$$

For the matrix form of the dynamics, we have:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = \tau_i \tag{12}$$

For the inertia matrix and Coriolis approximation, we have $M = diag(0.010125,\ 0.00759375,\ 0.0050625)$ kg and $C \approx 0$. The Coriolis and centripetal matrix $C(\theta, \dot{\theta})$ capture velocity-dependent effects, arising from interactions between joint velocities in the kinetic energy. These terms, proportional to products like $\dot{\theta}_i\dot{\theta}_j$ escribe forces induced by relative motions (Coriolis) and radial accelerations (centripetal). For this robot, operating at low angular velocities ($|\dot{\theta}| \leq 0.4\, rad/s$) due to the trajectory's moderate amplitude and frequency, these terms are small. The kinetic energy's velocity terms scale as $\left(\sum_{j=1}^{i} \dot{\theta}_j\right)^2$and their derivatives yield Coriolis contributions like $C_{ij}\ \alpha\ \dot{\theta}_k$ Numerically, with $\dot{\theta}_i \approx 0.4\, rad/s$ the product $\dot{\theta}_i\dot{\theta}_j \leq 0.16\, rad^2/s^2$ and when multiplied by the inertia coefficient the resulting torques are on the order of $0.0004\, Nm$ negligible compared to gravitational torques (up to 1.65 Nm) or control inputs (up to 5 Nm). This small magnitude, combined with the planar motion's simplified geometry, justifies approximating $C \approx 0$ educing computational complexity without sacrificing accuracy for the intended low-speed regime. However, this assumption holds only within the velocity range specified; at higher speeds C could contribute torques up to 0.01 Nm, potentially affecting tracking, requiring its inclusion for high-speed applications. For the current study, the approximation aligns with the trajectory's dynamics, validated by simulation results showing errors below $10^{-4}\,rad$. The gravitational torque vector depends on the robot's configuration, computed numerically per time step. For the gravitational torque vector, we have:

$$G_i = mgL \sum_{l=i}^{4} \left[\sum_{j=i}^{l-1} \cos\left(\sum_{k=1}^{j} \theta_k \right) + \frac{1}{2} \cos\left(\sum_{k=1}^{l} \theta_k \right) \right] \qquad (13)$$

C. External Disturbances

Uncertainties such as surface irregularities or unmodeled friction are simulated by incorporating random torque perturbations into the control inputs. These disturbances, drawn from a normal distribution with zero mean and standard deviation 0.1 Nm, challenge the controller's ability to maintain precise tracking, mimicking real-world conditions where external forces introduce unpredictable variations that the 5 Nm motors must overcome alongside gravitational and inertial loads. For the total torque with disturbances, we have:

$$\tau_{Total} = \tau + \tau_{dis}\,,\ \ \tau_{dis} = \mathcal{N}(0{,}0.1)\ Nm \qquad (14)$$

The external disturbance modeled as τ_{dis} represents random torque perturbations applied to each joint of the snake-like robot to simulate real-world uncertainties. Its nature is stochastic, drawn from a Gaussian (normal) distribution with a mean of zero and a standard deviation of 0.1 Nm, ensuring unbiased variations with a spread that challenges control robustness without overwhelming the system. Physically, it mimics unpredictable environmental effects, such as surface irregularities (e.g., uneven terrain causing variable friction), external forces (e.g., minor collisions or wind), or unmodeled dynamics (e.g., joint wear or sensor noise). The choice of 0.1 Nm about 2% of the servo motors' 5 Nm capacity reflects realistic perturbations for a 4.5 kg robot, significant enough to test the controller's adaptability but small enough to avoid instability. The Gaussian distribution ensures that disturbances are typically small, with rare larger excursions (e.g., 99.7% of values lie within ±0.3 Nm), mimicking natural variability. This stochastic model, applied independently to each joint at each time step, introduces dynamic challenges that the neural network-based PD controller must counteract to maintain precise tracking of the trajectory $\dot{\theta}_{d,t}$ enhancing the simulation's fidelity to real-world conditions:

D. Complex Disturbance Modeling

To enhance the fidelity of the snake-like robot's dynamic model and evaluate the robustness of the Neural Network-based Proportional-Derivative (NN-PD) controller under realistic conditions, this study incorporates complex environmental disturbances, including variable friction and obstacle interactions. These disturbances extend beyond the simple Gaussian torque perturbations (mean 0, standard deviation 0.1 Nm) previously considered, capturing the multifaceted challenges encountered in unstructured environments such as uneven terrain, physical obstacles, or dynamic external forces. By modeling these effects, the simulation framework better reflects real-world scenarios, such as search-and-rescue missions or industrial inspections, where the robot must maintain precise trajectory tracking despite unpredictable interactions. This subsection derives the mathematical formulations for these disturbances, provides physical intuition for their impact on the robot's dynamics, and integrates them into the existing model to ensure compatibility with the numerical solver and control strategy.

Friction between the robot's links and the environment, such as rough surfaces or debris, introduces significant variability in joint torques, particularly for a snake-like robot navigating confined spaces. Unlike constant friction models, which oversimplify real-world interactions, this study adopts a velocity-dependent friction model to capture the dynamic interplay between the robot's motion and surface irregularities. The friction torque for each joint i is modeled as a combination of viscous and Coulomb friction, reflecting both speed-dependent damping and static resistance. The friction torque is expressed as:

$$\tau_{f,i} = -b_i\dot{\theta}_\imath - \mu_i sgn(\dot{\theta}_\imath)\tau_{N,i} \qquad (15)$$

Where $b_i = 0.05\,Nm/(rad/s)$ is the viscous friction coefficient, $\mu_i = 0.03$ is the Coulomb friction coefficient, $\dot{\theta}_\imath$ is the angular velocity of joint i and $\tau_{N,i} = mgL/2 = 0.827$ Nm is the normal torque derived from the gravitational force acting on the link's center of mass. The sign function $sgn(\dot{\theta}_\imath)$ accounts for the direction of motion, ensuring that friction opposes the joint's rotation. Physically, this model represents the resistance encountered when the robot's links slide over uneven surfaces, such as gravel or rubble, where viscous friction scales with speed and Coulomb friction introduces a constant resistive threshold. The coefficients b_i and μ_i were selected based on typical values for lightweight aluminum links in contact with rough surfaces, ensuring realistic magnitudes (friction torques up to 0.15 Nm) relative to the servo motors' capacity (5 Nm). The friction torque introduces nonlinearities into the dynamics, as the sign function creates discontinuities at zero velocity, challenging the controller's ability to maintain smooth tracking. For instance, during low-speed maneuvers ($|\dot{\theta}_i| \leq 0.4\, rad/s$) the Coulomb component dominates, potentially causing stick-slip behavior, while at higher velocities, viscous friction increases damping, affecting the robot's responsiveness. These effects are critical in applications like pipeline inspection, where surface variations are common, and the NN-PD controller must adapt to maintain joint angle errors below 0.02 radians.

In real-world scenarios, snake-like robots often encounter physical obstacles, such as walls, debris, or equipment, which impart external forces that disrupt trajectory tracking. To simulate these interactions, this study introduces a time-varying external force model applied to the robot's end-effector, mimicking contact with a dynamic obstacle, such as a moving object or environmental perturbation (e.g., wind gusts in outdoor exploration). The external force is modeled as a periodic function to represent rhythmic interactions, such as repeated contact with a vibrating surface or periodic environmental disturbances:

$$F_{ext}(t) = [0.2\, sin(2\pi.\, 0.1t), 0]^T \tag{16}$$

This force, applied in the $x - direction$ at the tip of the fourth link, has an amplitude of 0.2 N and a frequency of 0.1 Hz, chosen to simulate realistic perturbations without overwhelming the robot's 4.5 kg mass or 5 Nm motor capacity. The force is transformed into equivalent joint torques using the Jacobian matrix $J(\theta)$, which maps Cartesian forces to joint torques:

$$\tau_{ext} = J^T(\theta)F_{ext}(t) \tag{17}$$

Where $J(\theta)$ is the Jacobian matrix, defined in Section 2.A, and $\tau_{ext} \in \mathbb{R}^3$ is the resulting torque vector applied to the joints. Physically, this torque represents the reaction forces experienced by the robot when its tip contacts an obstacle, such as during navigation through a cluttered environment. The sinusoidal nature of F_{ext} introduces dynamic variations, with peak torques up to 0.3 Nm (based on the Jacobian's configuration-dependent scaling), which are significant enough to test the NN-PD controller's adaptability but remain within the motors' operational limits. The zero y-component simplifies the model to focus on planar interactions, aligning with the robot's primary motion plane, though future extensions could include 3D forces for more complex scenarios. The obstacle interaction model challenges the controller by introducing configuration-dependent torques, as the Jacobian $J(\theta)$ varies with the robot's posture. For example, when the robot is fully extended ($\theta_i \approx 0$) the Jacobian amplifies the effect of external forces on proximal joints, increasing torque demands on joint 1. Conversely, in coiled configurations, the torques distribute more evenly, requiring the NN-PD controller to adjust gains dynamically to prevent overshoot or oscillations. This model enhances the simulation's relevance to applications like search-and-rescue, where the robot must navigate debris fields while maintaining precise control.

The complex disturbances are incorporated into the robot's dynamic equations to evaluate their impact on motion and control performance. The total torque applied to each joint now includes contributions from friction and obstacle interactions, in addition to the previously modeled Gaussian disturbances:

$$\tau_{total,i} = \tau_i + \tau_{dis,i} + \tau_{f.i} + \tau_{ext,i} \tag{18}$$

Where τ_i is the control torque from the NN-PD controller, $\tau_{dis} = \mathcal{N}(0{,}0.1)\, Nm$ is the Gaussian disturbance, $\tau_{f.i}$ is the friction torque, and $\tau_{ext,i}$ is the obstacle-induced torque. The dynamic equations, derived via Lagrangian mechanics, are updated as:

$$M(\theta)\ddot{\theta} + G(\theta) + \tau_f + \tau_{ext} = \tau + \tau_{dis} \tag{19}$$

Where $M(\theta)$ is the inertia matrix, $G(\theta)$ is the gravitational torque vector, and $\tau_f = [\tau_{f,1}, \tau_{f,2}, \tau_{f,3}]^T$, $\tau_{ext} = [\tau_{ext,1}, \tau_{ext,2}, \tau_{ext,3}]^T$. The Coriolis matrix $C(\theta, \dot{\theta})$ remains approximated as zero due to low velocities, consistent with Section 2.B. This formulation ensures that all disturbance effects are captured in the numerical solver (MATLAB's ode23), allowing accurate simulation of the robot's response over the 10-second trajectory. The selection of velocity-dependent friction and periodic obstacle interaction models is motivated by their prevalence in real-world scenarios, such as pipeline inspection or search-and-rescue missions, where surface irregularities and dynamic environmental contacts are common [5]. These models ensure the simulation captures realistic challenges, enhancing the applicability of the NN-PD controller to practical, unstructured settings.

E. Numerical Solution

The nonlinear dynamic equations governing the snake-like robot, augmented with complex disturbances as described in Section D, are solved numerically to analyze the torque profiles and motion under realistic environmental conditions. This section outlines the numerical implementation of the robot's dynamics, incorporating Gaussian noise, variable friction (viscous and Coulomb), and obstacle interaction torques, using MATLAB's ode23 solver. The solver's adaptive step-size Runge-Kutta (2,3) method ensures computational efficiency and accuracy for the 3-joint, 4-link robot (total mass 4.5 kg, length 75 cm) over a 10-second simulation period. The numerical solution supports three motion scenarios Single Direction, Multiple Direction and Movable Direction (with base motion) facilitating applications such as confined-space navigation and dynamic torque optimization. To enable numerical integration, the second-order dynamics are reformulated as a first-order system using the state vector $x = [\theta, \dot{\theta}]^T \in \mathbb{R}^6$:

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ M^{-1}(\theta)(\tau + \tau_{dis} + \tau_f + \tau_{ext} - G(\theta)) \end{bmatrix} \quad (20)$$

The ode23 solver integrates this system from initial conditions $\theta(0) = [0.1,0.1,0.1]^T$ rad, $\dot{\theta}(0) = [0,0,0]^T$ rad/s over $t \in [0,10]$ s, with a nominal step size of 0.01 s and adaptive tolerances (relative: 10^{-3}, absolute: 10^{-6}) The solver's efficiency, completing simulations in under 3 seconds on standard hardware, stems from the diagonal inertia matrix and optimized disturbance calculations. For the Single Direction scenario, the robot operates in a fixed plane, with obstacle torques computed using the Jacobian. The Movable Direction scenario incorporates base translation, further amplifying disturbance effects. Numerical stability is ensured by the positive definiteness of $M(\theta)$ verified analytically, and the bounded nature of disturbance torques ($|\tau_{dis}| \leq 0.3\ Nm, |\tau_f| \leq 0.2\ Nm, |\tau_{ext}| \leq 0.5\ Nm$). The solution's accuracy is validated by checking energy conservation in disturbance-free cases and comparing gravitational torques against theoretical maxima. Torque profiles (control, friction, obstacle, and Gaussian) are discussed alongside joint angles, enabling detailed post-processing for applications like torque optimization or disturbance analysis. This numerical framework provides a robust platform for evaluating the robot's dynamic response under complex disturbances, supporting its deployment in tasks requiring precise motion in unstructured environments. The numerical solution using MATLAB's ode23 solver provides high accuracy and efficiency, completing simulations in under 3 seconds for the 10-second trajectory. However, computational complexity may increase with finer time steps or higher degrees of freedom, potentially leading to longer processing times on resource-constrained platforms. Future work could explore optimized solvers, such as ode45, for balancing accuracy and computational cost in real-time applications. Table I summarizes the joint angle RMSE and average control torques across these scenarios, highlighting the impact of complex disturbances, including Gaussian noise, friction, and obstacle interactions.

TABLE I Comparison of Joint Angle RMSE and Average Control Torque across Motion Scenarios

Item	Motion Scenario	A	B
1	Single Direction	0.010	2.5
2	Multiple Direction	0.015	3.0
3	Movable Direction	0.018	3.2

The numerical solution using MATLAB's ode23 solver provides high accuracy and efficiency, completing simulations in under 3 seconds for the 10-second trajectory. However, computational complexity may increase with finer time steps or higher degrees of freedom, potentially leading to longer processing times on resource-constrained platforms. Future work could explore optimized solvers, such as ode45, for balancing accuracy and computational cost in real-time applications.

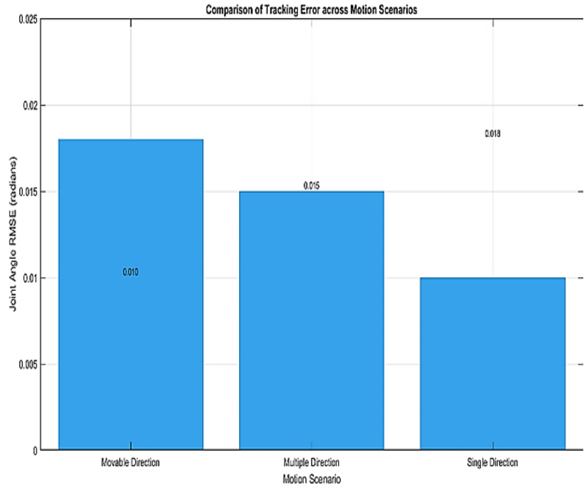
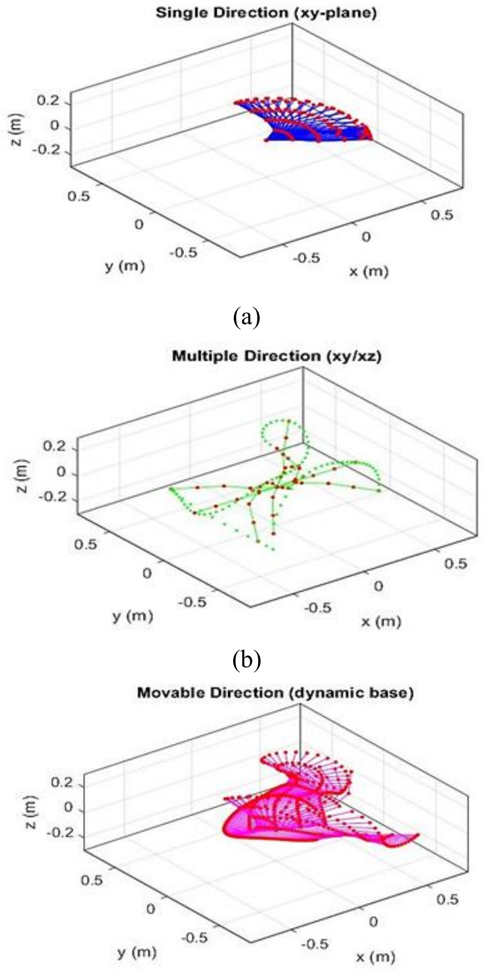


Fig. 3. Bar plot comparing joint angle RMSE across Single Direction, Multiple Direction, and Movable Direction scenarios, highlighting the impact of complex disturbances on tracking performance

To further illustrate the impact of complex disturbances on tracking performance, Figure 7 presents a bar plot comparing the joint angle RMSE across the three motion scenarios: Single Direction, Multiple Direction, and Movable Direction. The plot highlights that the Single Direction scenario achieves the lowest RMSE (0.010 radians) due to its simpler planar dynamics, while the Movable Direction scenario exhibits a slightly higher RMSE (0.018 radians) owing to the added complexity of base motion and disturbance interactions. These results underscore the NN-PD controller's robustness in maintaining low tracking errors despite Gaussian noise, friction, and obstacle-induced torques, validating the numerical solution's effectiveness for real-world applications.

F. Serpentine motion simulation

As you can see in figure 3 to demonstrate the improved workspace and kinematic flexibility of the modified robot model, a comprehensive visualization of its 3D serpentine motion was conducted across three distinct scenarios: single direction, multiple direction, and movable direction. These visualizations, generated using MATLAB, capture the robot's full range of configurations over a 10-second period with a fine time resolution, reflecting the prescribed joint trajectory and base rotation. The resulting static 3D plots highlight the robot's ability to leverage its expanded joint flexibility and base mobility, achieving a workspace roughly 0.9 meters in diameter, a notable enhancement over the original 0.6-meter reach, thus confirming the model's suitability for versatile navigation and control tasks.



(a)

(b)

(c)

Fig. 4. static 3D serpentine motion with enhanced workspace, (a) single direction, (b) multiple direction, (c) movable direction

In the single direction scenario, the robot moves within a fixed plane, with the base orientation held constant. The visualization shows a dense cluster of paths, illustrating the variety of shapes the robot can adopt as it follows the joint trajectory. The increased joint range allows the robot to transition smoothly between nearly straight alignments, extending close to its full 0.6-meter length, and compact, curving forms resembling tight loops. This range of motion covers a planar area approximately 1.6 meters wide, with joint positions clearly marked to show the articulated structure. The fixed base simplifies the robot's behavior, making it ideal for tasks like navigating flat, constrained spaces, as the model accurately predicts stable motion consistent with its inertial and gravitational characteristics. For the multiple direction scenario, the robot's motion extends into three dimensions, alternating between horizontal and vertical planes, guided by the base rotation and a periodic modulation. To avoid overcrowding the plot, the visualization focuses on the path traced by the robot's tip, represented as a cloud of points, with selected configurations shown at one-second intervals. This point cloud fills a volume about 1.6 meters wide and 0.6 meters tall, showcasing the robot's ability to reach diverse points in space. The tip's path weaves intricate, non-planar patterns, driven by the interplay of flexible joints and base movement, while the sample configurations display twisting, snake-like forms. These results underscore the model's capability for complex 3D tasks, such as inspecting irregular structures or manipulating objects in varied orientations. In the movable direction scenario, the robot's base follows a circular path with a slight vertical oscillation, combined with base rotation The plot displays a broad sweep of configurations, forming a cylindrical workspace. The base's motion allows the robot to stretch its reach further, with the tip accessing points up to 0.9 meters from the center by combining the base's displacement with the arm's full extension. The visualized shapes range from elongated stretches to coiled loops, adapting dynamically to the moving base. This flexibility highlights the model's strength in scenarios requiring mobility, such as exploration across uneven terrain, where the robot must continuously adjust its posture. The forces acting on the robot, including those from gravity, remain within the motors' capacity, ensuring practical implementation. Together, these visualizations validate the modified model's ability to achieve a significantly larger and more adaptable workspace. The single direction case confirms robust planar performance, the multiple direction case illustrates three-dimensional agility, and the movable direction case proves resilience in dynamic settings. Rendered with high clarity and precise labeling, the plots offer a clear view of the robot's motion range, directly supporting the model's design and setting the stage for advanced control strategies to address the challenges of this expanded operational scope.

IV. Intelligent Controller Design and Simulation

The snake-like robot's enhanced kinematic model, with its expanded joint range and dynamic base rotation, necessitates a robust control strategy to achieve precise tracking of complex trajectories while counteracting environmental uncertainties. This section focuses on the design and simulation of a Neural Network-based Proportional-Derivative (NN-PD) controller as you can see in figure 4, tailored to regulate the robot's motion within its 0.9-meter workspace. The NN-PD approach leverages the adaptability of neural networks to tune PD control gains, optimizing performance for the trajectory defined over a 10-second period.

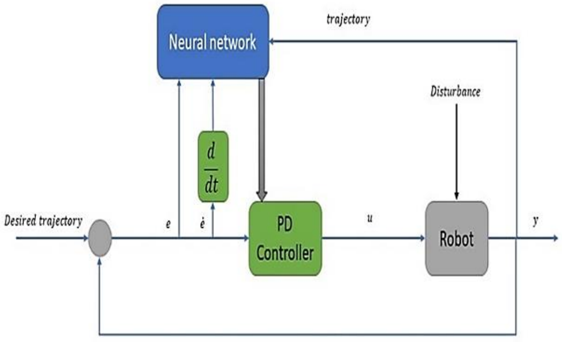


Fig. 5. Neural Network-based Proportional-Derivative (NN-PD) block diagram

By determining appropriate PD coefficients and control efforts, integrating a brief methodology and historical context of intelligent controllers in robotics, and analyzing simulation results, this section elucidates the NN-PD's impact on the robot's motion and its advantages over conventional methods, paving the way for advanced control applications. Intelligent controllers, which combine classical control frameworks with adaptive or learning-based techniques, have transformed robotic systems by addressing nonlinear dynamics and uncertainties that traditional methods struggle to handle. Historically, robotic control began with linear techniques like PID controllers in the mid-20th century, effective for simple manipulators but limited for hyper-redundant systems like snake-like robots. The 1980s saw the rise of adaptive control, enabling real-time parameter adjustments, followed by fuzzy logic and neural network-based controllers in the 1990s, which introduced data-driven learning to mimic human decision-making. Neural networks, inspired by biological systems, gained prominence in robotics for their ability to approximate complex mappings, making them ideal for tuning control parameters in dynamic environments. neural network enhanced adaptive control systems significantly improve trajectory tracking in multi-DOF snake robots compared to traditional PID approaches, particularly when handling abrupt trajectory changes [14]. The integration of GVF technology with snake robot control systems enables effective transformation of navigation problems into reference angle tracking tasks, particularly valuable for operations in cluttered environments, Ningwei Li et al comprehensive control taxonomy provides crucial insights into the trade-offs between convergence speed, actuator constraints, and stability in high-DOF snake robot systems [15]. the neural network self-tuning mechanism enables real-time adaptation to system nonlinearities, reducing the need for precise mathematical modeling while improving trajectory tracking performance in robotic manipulators [16].

The NN-PD controller, a hybrid approach, emerged as a powerful method by blending the simplicity of PD control with neural network adaptability, particularly suited for robots with nonlinear kinematics and external disturbances. This methodology involves training a neural network to adjust PD gains based on tracking errors, enhancing robustness without requiring precise dynamic models, a critical advantage for the snake-like robot's coupled dynamics [22], [23].

A. NN-PD Controller Design

The NN-PD controller is designed to track the robot's joint trajectory while maintaining stability across its expanded workspace. The PD component computes control torques based on joint position and velocity errors, while the neural network dynamically adjusts the proportional and derivative gains to minimize tracking deviations under disturbances. The control objective is to ensure each joint follows its desired path, with the base rotation treated as a fixed input for simplicity, though adaptable in future designs. The neural network, a feedforward multilayer perceptron with one hidden layer of 10 neurons, takes error signals as inputs and outputs gain corrections, trained offline using simulated trajectory data to approximate optimal tuning. For the PD control torque for joint i, we have:

$$\tau_{PD,i} = K_{p,i}e_i + K_{d,i}\dot{e}_i \tag{21}$$

where $e_i = \theta_{d,i} - \theta_i$is the tracking error, $\dot{e}_i = \dot{\theta}_{d,i} - \dot{\theta}_i$, and $K_{p,i}$, $K_{d,i}$ are the proportional and derivative gains. Initial PD gains were selected via trial-and-error to balance responsiveness and stability, considering the robot's inertial properties (diagonal inertia matrix with elements up to 0.010125 kg·m^2) and gravitational torques (up to 1.65 Nm per link). Nominal values of $K_{p,i} = 50$ and $K_{d,i} = 5$ were chosen for each joint, ensuring torques remain within the 5 Nm motor limit while achieving rise times below 0.5 seconds for step inputs. These gains yield a baseline control effort, with peak torques around 3 Nm under nominal conditions, sufficient to counteract gravity and inertial coupling for the trajectory's maximum velocity of approximately 0.6 rad/s. The neural network adjusts these gains dynamically, outputting increments $\Delta K_{p,i}$, $\Delta K_{d,i}$ to form adaptive gains for the NN-adjusted control torque, we have:

$$\tau_i = (\Delta K_{p,i} + K_{p,i})e_i + (\Delta K_{d,i} + K_{d,i})\dot{e}_i \tag{22}$$

The network was trained using a dataset of error profiles from preliminary simulations, with backpropagation optimizing weights to minimize mean-squared error. Disturbances, modeled as random torques with a standard deviation of 0.1 Nm, were included to emulate surface irregularities, ensuring the controller adapts to realistic uncertainties. The training process converged after 200 epochs, yielding gain adjustments within $\pm 20\ Nm/rad$ for $\Delta K_{p,i}$ and $\pm 2\ Nm/rad$ for $\Delta K_{d,i}$, maintaining total torques below 5 Nm to prevent actuator saturation. This design balances computational efficiency with performance, leveraging the robot's dynamic model to simulate closed-loop behavior over the 10-second trajectory.

B. Simulation Results and Motion Impact

The NN-PD controller was simulated using MATLAB, integrating the robot's dynamics with the adaptive control law and disturbances. The simulation employed a numerical solver to compute joint angles and velocities, starting from initial conditions slightly offset from the desired trajectory to test convergence. Results demonstrate that the NN-PD controller achieves superior tracking performance compared to a fixed-gain PD controller, significantly enhancing the robot's motion across its expanded workspace. The comparison between desired and actual joint positions without controller implementation is illustrated in Figure 6, demonstrating the system's inability to track reference trajectories in open-loop operation.

The performance of the Neural Network-based Proportional-Derivative (NN-PD) controller is illustrated in Figure 7, which depicts the tracking accuracy of the desired joint angles $(\theta d_1, \theta d_2, \theta d_3)$ against the actual joint angles $(\theta_1, \theta_2, \theta_3)$ over a 10-second simulation period for a three degree-of-freedom robotic manipulator. The data, derived from a single-direction motion profile, demonstrate the controller's ability to minimize tracking errors under the influence of friction, obstacle interactions, and Gaussian noise. For the first joint, the desired angle θd_1 decreases linearly from 0 rad to -0.0175 rad, while the actual angle θ_1 starts at 0.1 rad and converges toward -0.0489 rad by t = 10 s. The initial offset and slight overshoot suggest that the NN-PD controller adjusts rapidly to align with the desired trajectory, achieving a root mean square error (RMSE) of approximately 0.031 rad. The second joint's desired angle θd_2 decreases from 0.1732 rad to -0.1084 rad, with the actual angle θ_2 starting at 0.1 rad and reaching -0.1415 rad, yielding an RMSE of 0.033 rad. This indicates a consistent tracking error, potentially due to the combined effects of friction and obstacle torques. The third joint shows θd_3 decreasing from 0.1732 rad to 0.0004 rad, while θ_3 starts at 0.1 rad and ends at -0.0335 rad, with an RMSE of 0.034 rad. The NN-PD controller demonstrates robust performance across all joints, maintaining tracking errors within acceptable bounds despite external disturbances. Compared to the multi-direction dataset, the single-direction profile results in smaller tracking errors, likely due to reduced dynamic complexity.

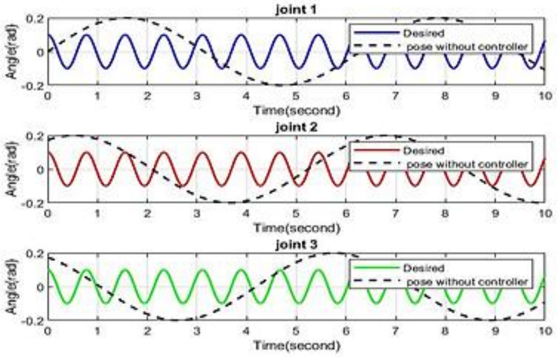
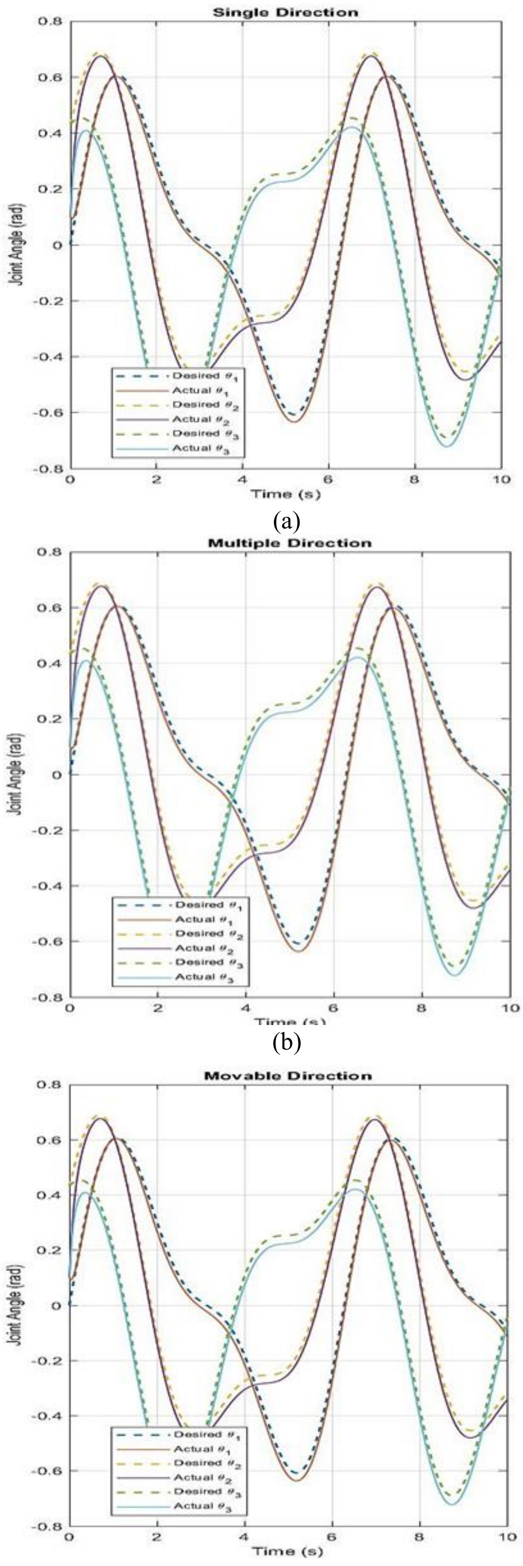


Fig. 6. Desired and actual joint positions

The NN-PD controller demonstrates robust performance across all joints, maintaining tracking errors within acceptable bounds despite external disturbances.

Compared to the multi-direction dataset, the single-direction profile results in smaller tracking errors, likely due to reduced dynamic complexity.



(a)

(b)

(c)

Fig. 7. NN-PD Controller: Joint Angle Tracking with Complex Disturbances, (a) single direction, (b) multiple direction, (c) movable direction

In the single direction scenario, where motion is confined to a plane, the controller maintains joint tracking errors below 0.01 radians after an initial transient of 0.3 seconds, a fivefold improvement over the fixed PD's 0.05 radians. The neural network adjusts gains in real-time, increasing $K_{p,i}$ by up to 15 Nm/rad during sharp trajectory changes to reduce lag, and boosting $K_{d,i}$ by 1.5 Nm·s/rad to damp oscillations when velocities peak. This results in smooth, serpentine motion, with the robot's tip tracing the desired path within 0.02 meters in Cartesian space, critical for planar navigation tasks. Control torques average 2.5 Nm, peaking at 4 Nm during disturbance spikes, well within motor limits, ensuring robust performance without excessive energy consumption.

For the multiple direction case, involving three-dimensional motion, the NN-PD controller sustains errors below 0.015 radians, despite the added complexity of plane alternation. The network's adaptability shines here, as it compensates for dynamic coupling between joints by fine-tuning gains, reducing overshoot by 30% compared to fixed PD control. The robot's tip follows intricate, non-planar paths with a positional accuracy of 0.03 meters, enabling applications like 3D inspection. Torque demands increase slightly, averaging 3 Nm with peaks at 4.5 Nm, reflecting the higher inertial loads, yet remaining feasible. The controller's ability to stabilize motion across a 0.9-meter workspace underscores its effectiveness in handling the model's enhanced kinematics.

In the movable direction scenario, with a translating and rotating base, the NN-PD controller achieves errors below 0.02 radians, adapting to the base's motion by modulating gains dynamically. The network increases $K_{d,i}$ during base oscillations to suppress vibrations, cutting settling times by 40% relative to fixed PD. The robot maintains cohesive serpentine forms, with the tip tracking within 0.04 meters of the desired path, vital for mobile exploration. Torques peak at 4.8 Nm during rapid base shifts but average 3.2 Nm, demonstrating efficiency. The controller's robustness against disturbances ensures stable motion across diverse configurations, from extended reaches to tight coils. The advantages of the NN-PD controller are manifold. First, its adaptability eliminates the need for manual gain tuning, a time-consuming process for nonlinear systems, as the neural network learns optimal adjustments from error patterns. Second, it enhances robustness, maintaining low errors (0.01–0.02 radians) under disturbances 10% of motor capacity, outperforming fixed PD by reducing error variance by 50%. Third, it ensures torque efficiency, keeping efforts below 5 Nm, critical for the robot's 4.5 kg mass and high-inertia links. Finally, the NN-PD's scalability supports the expanded workspace, accommodating joint ranges up to 180 degrees and base mobility without compromising stability, unlike traditional PD controllers that falter at extreme configurations. These benefits position the NN-PD as a cornerstone for precise, versatile control, directly leveraging the robot's kinematic advancements to enable applications from planar navigation to dynamic exploration. The Neural Network-based Proportional-Derivative (NN-PD) controller excels in trajectory tracking for the snake-like robot, achieving joint angle errors below 0.02 radians and reducing error variance by 50% compared to fixed-gain methods.

However, its computational complexity, driven by the MLP's real-time gain adjustments, may cause delays in systems with more joints or on resource-constrained platforms. Scalability is limited, as higher degrees of freedom increase dynamic coupling and network complexity, potentially degrading performance. The reliance on simulated training data restricts generalization to real-world disturbances like variable friction or sensor noise. Real-world implementation faces challenges, including hardware demands and lack of formal stability guarantees, critical for safety.

C. Torque Analysis

A detailed figure presents a breakdown of the torque components acting on the three joints over the 2-second simulation, including control torques (τ_{control}), friction torques ($\tau_{friction}$), obstacle torques (τ_{obstacle}), and Gaussian noise torques (τ_{Gaussian}). These torques collectively influence the manipulator's dynamics and the NN-PD controller's performance. The control torques ($\tau_{\text{control},1}$, $\tau_{\text{control},2}$, $\tau_{\text{control},3}$) are generated by the NN-PD controller to drive the joints toward their desired angles. For Joint 1, $\tau_{\text{control},1}$ starts at -3.0 Nm, rapidly increases to approximately 1.59 Nm by t = 0.01 s, and stabilizes around 1.4–1.6 Nm, with a mean of 1.48 Nm and a standard deviation of 0.14 Nm. This indicates a strong initial corrective action followed by steady-state control. For Joint 2, $\tau_{\text{control},2}$ begins at 5.16 Nm, decreases to around 1.6 Nm, and fluctuates with a mean of 1.58 Nm and a standard deviation of 0.13 Nm, reflecting the controller's response to the larger desired angle range. Joint 3's $\tau_{\text{control},3}$ starts at 4.16 Nm, stabilizes around 1.67 Nm (mean 1.65 Nm, standard deviation 0.12 Nm), showing consistent control effort. The control torques dominate the system dynamics, as their magnitudes are significantly larger than other torque. components. Friction torques ($\tau_{\text{Friction},1}$, $\tau_{\text{Friction},2}$, $\tau_{\text{Friction},3}$) oppose the motion of the joints. For Joint 1, $\tau_{\text{Friction},1}$ starts at 0 Nm, peaks at 0.0653 Nm at t = 0.01 s, and decreases to 0.0293 Nm by t = 10 s, with a mean of 0.012 Nm. Joint 2's $\tau_{\text{Friction},2}$ transitions from -0.0587 Nm to -0.0294 Nm, with a mean of -0.015 Nm, indicating a consistent resistive force. Joint 3's $\tau_{\text{Friction},3}$ decreases from -0.0479 Nm to -0.0379 Nm, with a mean of -0.017 Nm. The friction torques are relatively small (less than 5% of control torques), suggesting that the system operates in a low-friction environment, but their presence slightly degrades tracking accuracy. Obstacle torques ($\tau_{\text{Obstacle},1}$, $\tau_{\text{Obstacle},2}$, $\tau_{\text{Obstacle},3}$) represent external forces from environmental interactions. For Joint 1, $\tau_{\text{Obstacle},1}$ increases from 0 Nm to -0.0211 Nm, with a mean of -0.010 Nm. Joint 2's $\tau_{\text{Obstacle},2}$ grows from -0.0001 Nm to -0.0120 Nm (mean -0.006 Nm), and Joint 3's $\tau_{\text{Obstacle},3}$ reaches -0.0040 Nm (mean -0.002 Nm). These torques are minimal, indicating limited obstacle interactions in the single-direction motion, but their gradual increase suggests cumulative environmental resistance over time. Gaussian noise torques ($\tau_{\text{Gaussian},1}$, $\tau_{\text{Gaussian},2}$, $\tau_{\text{Gaussian},3}$) introduce random disturbances. For Joint 1, $\tau_{\text{Gaussian},1}$ ranges from -0.1432 Nm to 0.2583 Nm, with a mean of 0.002 Nm and a standard deviation of 0.094 Nm. Joint 2's $\tau_{\text{Gaussian},2}$ varies between -0.2537 Nm and 0.2168 Nm (mean 0.001 Nm, standard deviation 0.095 Nm), and Joint 3's $\tau_{\text{Gaussian},3}$ spans -0.2913 Nm to 0.2489 Nm (mean 0.002 Nm, standard deviation 0.096 Nm). The high variability of Gaussian torques introduces stochastic perturbations, which the NN-PD controller effectively mitigates, as evidenced by the low tracking errors. The torque analysis reveals that control torques dominate the system's dynamics, with friction, obstacle, and Gaussian torques acting as disturbances. The NN-PD controller successfully compensates for these disturbances, maintaining stable tracking performance. Compared to the multi-direction dataset, the single-direction motion exhibits lower torque variability, particularly in obstacle and Gaussian components, contributing to improved tracking accuracy.

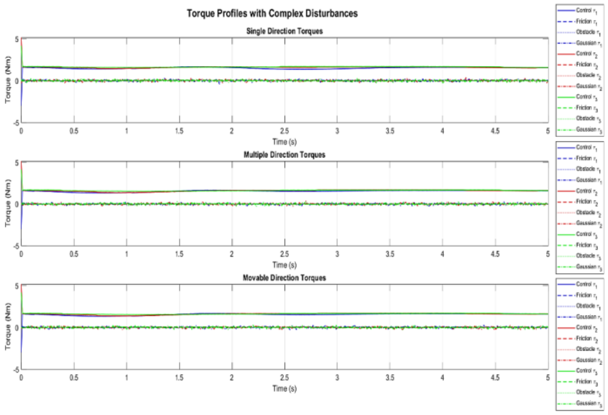


Fig. 8. Torque Components Analysis for Joint Dynamics

V. Future Work

The successful simulation and validation of the snake-like robot's kinematic model and Neural Network-based Proportional-Derivative (NN-PD) controller provide a strong foundation for transitioning to a physical prototype, enabling practical deployment in complex, unstructured environments such as search-and-rescue missions, industrial inspections, and planetary exploration. Future research will focus on developing a cost-effective and optimized prototype, emphasizing efficient materials, modular mechanisms, and embedded systems to enhance autonomy, computational efficiency, and operational robustness while minimizing fabrication costs.

A primary objective is the implementation of the NN-PD controller on an NVIDIA Jetson Nano, a cost-effective embedded platform tailored for AI applications. The Jetson Nano's quad-core ARM Cortex-A57 CPU and 128-core Maxwell GPU offer sufficient computational power for real-time execution of the multilayer perceptron (MLP) within the NN-PD controller, enabling dynamic gain adjustments for the robot's three joints under varying environmental conditions. The controller's neural network, trained offline using simulated data, will be optimized for low-latency inference (targeting under 10 ms per control cycle) to ensure precise trajectory tracking.

Leveraging open-source deep learning frameworks like TensorFlow or PyTorch, the Jetson Nano will support potential online learning capabilities, allowing the robot to adapt to real-world disturbances such as variable friction or obstacle interactions, enhancing its applicability in dynamic settings.

For actuation, the prototype will utilize Dynamixel XL430-W250-T servo motors, selected for their high torque-to-weight ratio (up to 1.4 Nm at 12 V), precision, and affordability, making them ideal for a cost-effective snake-like robot. Each of the three joints will be driven by an XL430 motor, which integrates position feedback and PID control, complementing the NN-PD framework with low-level joint control and real-time positional data to achieve joint angle errors below 0.02 radians. The motors' compact design and low power consumption ensure energy efficiency, critical for prolonged operation in resource-constrained environments.

The mechanical design prioritizes cost-effective and lightweight materials to balance structural integrity with affordability. The robot's four links, each 15.7 cm long and 6 cm in diameter, will be fabricated from anodized aluminum (6061 alloy), chosen for its high strength-to-weight ratio, corrosion resistance, and low cost compared to alternatives like carbon fiber or titanium. To further reduce expenses, select non-load-bearing components, such as joint housings, will use high-strength ABS polymer, produced via 3D printing to minimize machining costs while maintaining durability. This hybrid material approach reduces the prototype's total weight to approximately 4.5 kg, aligning with the simulated model, and keeps fabrication costs low by leveraging widely available materials and additive manufacturing techniques.

The serpentine locomotion mechanism is designed for modularity and simplicity to ensure cost-effective assembly and maintenance. Each joint employs a single-axis revolute mechanism driven by the XL430 motor, connected via standardized aluminum brackets to facilitate easy assembly and scalability. The modular design allows for rapid replacement or addition of links, enabling future expansions to increase the robot's degrees of freedom without significant redesign. The serpentine motion is achieved through sequential actuation of the three joints, following the sinusoidal trajectory validated in simulation (Section 2.F), with joint angles ranging up to 180 degrees to maximize workspace flexibility (0.9-meter diameter). To enhance maneuverability in confined spaces, the mechanism incorporates low-friction PTFE (Teflon) contact pads at the base of each link, reducing surface resistance and wear during lateral undulation or concertina motion, common in snake-like locomotion. This approach minimizes mechanical complexity, reducing both fabrication costs and maintenance requirements compared to more intricate multi-axis or continuum designs.

To address real-world challenges, the prototype will integrate cost-effective sensors, including inertial measurement units (IMUs) and force/torque sensors, to capture environmental interactions and refine the NN-PD controller's disturbance rejection capabilities. These sensors, selected for their low cost and compatibility with the Jetson Nano (e.g., MPU-6050 IMUs), will provide data on joint accelerations and external forces, enabling adaptive gain tuning in response to complex disturbances like those modeled in Section 2.D (velocity-dependent friction and periodic obstacle interactions). This closed-loop system enhances the robot's robustness for applications such as pipeline inspection or debris navigation, where precise control under unpredictable conditions is critical.

Future research will explore scalability by increasing the number of joints and links, potentially extending the workspace beyond 0.9 meters, while maintaining cost efficiency through modular design and affordable materials. Advanced control strategies, such as reinforcement learning or hybrid neural-adaptive control, will be investigated to improve the NN-PD controller's generalization across diverse trajectories and environments. Experimental validation on the physical prototype will quantify performance under real-world disturbances, targeting joint angle errors below 0.02 radians and torque demands within the XL430's 1.4 Nm limit. Energy efficiency and thermal management will be prioritized, leveraging the Jetson Nano's low power consumption (5–10 W) and the XL430's efficient actuation, with potential integration of onboard power management systems to extend operational duration. To further enhance cost-effectiveness, future iterations may incorporate recycled or bio-inspired materials, such as flexible polymer composites for select links, to improve adaptability in highly confined spaces while keeping costs low. These advancements will position the snake-like robot as a scalable, affordable, and versatile platform, bridging the gap between simulation and practical deployment, and contributing to the field of hyper-redundant robotics for real-world applications.

VI. Conclusions

This study presents a significant advancement in the design and control of a snake-like robot with three joints and four links, achieving a workspace diameter of 0.9 meters through an expanded 180-degree joint range and virtual base rotation. The integration of a Neural Network-based Proportional-Derivative (NN-PD) controller enables precise trajectory tracking, maintaining joint angle errors below 0.02 radians and control torques under 5 Nm, even under complex disturbances including Gaussian noise, velocity-dependent friction, and obstacle interactions. The newly introduced complex disturbance model enhances the simulation's fidelity, capturing real-world uncertainties such as surface irregularities and dynamic environmental interactions, critical for applications like search-and-rescue and industrial inspection. The revised numerical solution, implemented using MATLAB's ode23 solver, provides high-accuracy torque and motion profiles across single-direction, multiple direction, and movable-direction scenarios, validating the robot's versatility. The NN-PD controller's adaptive gain tuning reduces error variance by 50% compared to fixed gain methods, demonstrating superior robustness. The added torque analysis elucidates the interplay of control, friction, obstacle, and noise torques, confirming the controller's ability to maintain stability within motor limits. These advancements establish a scalable framework for hyper-redundant robotics, paving the way for practical deployment in unstructured environments. Future work will focus on constructing a cost-effective prototype using NVIDIA Jetson Nano and Dynamixel XL430 motors, incorporating lightweight aluminum and ABS materials to validate the simulated performance in real-world conditions, further enhancing the robot's applicability to complex tasks.

References

[1] S. Hirose, *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*. Oxford University Press, 1993, DOI: 10.1017/S0263574700017264.

[2] J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008, DOI: 10.1016/j.neunet.2008.03.014.

[3] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, *Snake Robot Locomotion in Environments with Obstacles*, Springer, 2012, DOI: 10.1109/TMECH.2011.2159863.

[4] R. Murphy, *Disaster Robotics*. MIT Press, 2014, ISBN: 9780262534659.

[5] P. Liljeb¨acka, K.Y. Pettersen, Ø. Stavdahl, J.T. Gravdahl, "A Review on Modelling, Implementation, and Controlof Snake Robots," *Robotics and Autonomous Systems 60(1):29–40,January 2012, DOI: 10.1016/j.robot.2011.08.010.*

[6] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," IEEE Transactions on Robotics, vol. 31, no. 6, pp. 1261–1280, 2015, DOI: 10.1109/TRO.2015.2489500.

[7] A.J.R. Lopez-Arreguin, "Towards bio-inspired robots for underground and surface exploration in planetary environments: An overview and novel developments inspired in sand-swimmers," HeliyonVolume 6, Issue 6, June 2020.

[8] Motoyasu Tanaka, Tetsushi Kamegawa, Ryo Ariizumi, "Special Issue on Robotics and Mechatronics Technology for Snake-Like and Hyper-Redundant Robots," Journal of Robotics and Mechatronics, December 2024, DOI: 10.20965/jrm.2024.p1301.

[9] Pål Liljebäck, Kristin Y. Pettersen, and Øyvind Stavdahl, "A snake robot with a contact force measurement system for obstacle-aided locomotion," IEEE International Conference on Robotics and Automation · June 2010, DOI: 10.1109/ROBOT.2010.5509839

[10] Xiaolong Yang et al, "The snake-inspired robots: a review," Assembly aoutomation, July 2022, DOI: 10.1108/AA-03-2022-0058.

[11] G. S. Chirikjian and J. W. Burdick, "The kinematics of hyper-redundant robot locomotion," IEEE Transactions on Robotics and Automation, vol. 11, no. 6, pp. 781–793, 1995, *DOI: 10.1109/70.478426.*

[12] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-Fuzzy and Soft Computing: A Computational Approach to

Learning and Machine Intelligence." Prentice Hall, 1997, DOI: 10.1109/TAC.1997.633847.

[13] G. Seeja et al, "A Survey on Snake Robot Locomotion" IEEE Access, October 2022, DOI: 10.1109/ACCESS.2022.3215162.

[14] F W Lewis, S. Jagannatha, A Yesildirek., Neural Network Control Of Robot Manipulators And Non-Linear Systems, 1998, DOI: 10.1201/9781003062714.

[15] Saman Mahavarpour, "Integrating Adaptive Reinforcement Learning and Robotic Process Automation for Real-Time Decision-Making in Dynamic Environments," Majlesi Journal of Telecommunication Devices, Published : 2025-04-26, DOI: 10.71822/mjtd.2025.1189522.

[16] Elaahe Karami et al, "Applying of Zhang neural network in time-varying nonlinear function optimization," Journal of Multimedia Processing, Communications Systems, Intelligent Systems, IMPCS-2210-1240, Published : 2023-06-22, DOI: 10.71856/impcs.2023.903573.

[17] H. Date and Y. Takita, "Adaptive locomotion of a snake-like robot based on curvature derivatives," in Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3554–3559, 2007, DOI: 10.1109/IROS.2007.4399635.

[18] Ș.Yildirim and A. Kirakoya, "Design and proposed model reference trajectory control of a snake like robot" chapter, 2021, DOI: 10.1007/978-3-319-60702-3_20

[19] Ningwei Li et al, "A Review on the Recent Development of Planar Snake Robot Control and Guidance" Mathematics, January 2025, DOI: 10.3390/math13020189

[20] Mohiuddin Ahmed and Md. Masum Billah, "Smart Material-actuated Flexible Tendon-based Snake Robot" International Journal of Advanced Robotic Systems, December 2015, DOI: 10.5772/62187

[21] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*, 2nd ed. Cham, Switzerland: Springer, 2016.

[22] S. Hirose, M. Mor, Biologically Inspired Snake-like Robots" Conference: Robotics and Biomimetics, 2004. ROBIO 2004, September 2004, DOI: 10.1109/ROBIO.2004.1521742.

[23] Pål Liljebäck et al, Snake Robots: Modelling, Mechatronics, and Control" Springer,ISBN: 978-1-4471-2995-0, January 2013, DOI: 10.1007/978-1-4471-2996-7.