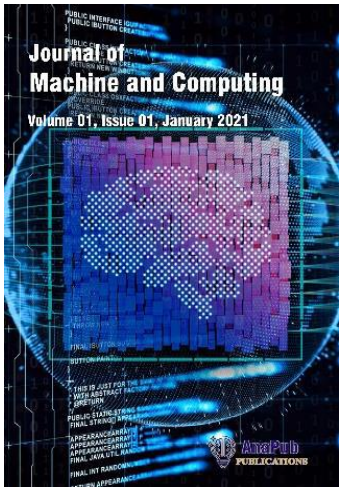
Journal Pre-proof

Bio-Inspired Adaptive Control Mechanisms in Mechatronic Systems Using Multi-Objective Evolutionary Deep Learning Optimization Techniques



Vishnukumar A, Manigandan S K, Ramya D, Revathy P and Balamurugan K

DOI: 10.53759/7669/jmc202505193
Reference: JMC202505193
Journal: Journal of Machine and Computing.

Received 18 April 2025
Revised from 29 June 2025
Accepted 05 August 2025

Please cite this article as: Vishnukumar A, Manigandan S K, Ramya D, Revathy P and Balamurugan K, "Bio-Inspired Adaptive Control Mechanisms in Mechatronic Systems Using Multi-Objective Evolutionary Deep Learning Optimization Techniques", Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505193.

AnaPub
PUBLICATIONS

Bio-Inspired Adaptive Control Mechanisms in Mechatronic Systems Using Multi-Objective Evolutionary Deep Learning Optimization Techniques

[1]Vishnukumar A, [2]Manigandan S K, [3]Ramya D, [4]Revathy P, [5]Balamurugan K [*]

[1]Department of Computer Science and Business Systems
Rajalakshmi Engineering College, Chennai – 602105

[2]Department of Information Technology
Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai

[3]Department of Information Technology
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai

[4]Department of Computer Science and Engineering
Thamirabharani Engineering College, Tirunelveli

[5]Department of Electronics and Communication Engineering
Saveetha School of Engineering
Saveetha Institute of Medical and Technical Sciences
Saveetha University, Chennai-602105

[1]vishnukumar1612@gmail.com, [2]manigandan@gmail.com, [3]raminfo84@gmail.com,
[4]revathypetchiappan@gmail.com, [5]bala237115@gmail.com

Corresponding Author: **Balamurugan K**

Abstract -The use of robotic arms in mechatronic systems is quite common because of their precision and adaptability uses, but the control of such nonlinear and dynamic systems has been an uphill task because of the presence of uncertainties and external disturbances. In this regard, the proposed study will solve the mentioned problems by designing an effective adaptive control approach to improve the accuracy of trajectory tracking, the system energy consumption and stability. The novelty of this study is to incorporate Echo State Network (ESN) with a hybrid met heuristic algorithm, which consists of Harris Hawks Optimisation (HHO) and Reptile Search Algorithm (RSA) to tune the important parameters of ESN, such as spectral radius, leakage rate and scaling of input. The described ESN-RSA-HHO framework will have a closed-loop architecture that will produce optimised torque commands to provide robust control of a 2-DOF robotic arm that operates under different operation conditions. Simulation has revealed that the ESN-RSA-HHO controller produces a root mean square tracking error of 0.012 rad, an energy saving of 28 per cent and an overshoot of 2.8 per cent, which is entirely better than the traditional PID control and LSTM-based control, as well as the non-optimised ESN models. The convergence behaviour and phase plane plot prove that the system can continue to be stable in even disturbed cases. The results confirm the efficiency of the suggested adaptive robot control framework and allow noting its future use in the mechatronic sphere.

Keywords: Adaptive Mechanism, Echo State Network, Harris Hawks Optimisation, Reptile Search Algorithm, 2-DOF Robotic Arms,

I. INTRODUCTION

Robotic systems have become indispensable in a wide spectrum of industries, revolutionising the ways of manufacturing, healthcare, aerospace and logistics. In manufacturing, robot arms are employed for assembly, welding, painting, and material handling, which enables high-speed and high-collective operations that exceed human abilities [1]. In healthcare, robots assist delicate surgical processes, rehabilitation remedies and laboratory

automation, where accuracy and stability are important. The aerospace industry depends on robotic systems to collect complex components, to conduct maintenance in dangerous environments, and even to assist in space exploration missions [2]. Similarly, logistics and warehousing sectors efficiently benefit from robotic systems for sorting, packaging and transportation. The root of these applications is the robotic arm, a highly versatile manipulator designed to perform complex, repetitive or dangerous tasks [3]. These weapons require an accurate and strong control mechanism to achieve high levels of accuracy, mastery and adaptability. These mechanisms should ensure smooth trajectory tracking, manage dynamic interaction with the environment, and maintain stability in external disturbances or the presence of system uncertainty. - As tasks become rapidly sophisticated – such as handling variable payloads or cooperating safely with humans – traditional control methods often decrease. This growing demand for autonomous and adaptive robotic weapons has catalysed research in advanced control strategies that are capable of meeting the challenges generated by non-linear dynamics, high degrees of independence and multi-purpose performance requirements [4].

Traditional control strategies such as proportional-integral-derivative (PID) and model predictive control (MPC) have long been the cornerstone of robotic arm control due to their simplicity, ease of implementation and effectiveness in structured, linear environments [5]. PID controllers, in particular, are widely used for tracking and status control because they provide a direct way to correct errors between desired and real positions. Similarly, the MPC provides benefits in involving obstacles by handling multi-perceived systems and predicting the behaviour of the future system on a finite time horizon [6]. Tuning PID benefits for optimal performance is also nontrivial and cannot be normalised in various functions or environments. MPC, while more sophisticated, suffers from computational overheads due to real-time adaptation requirements, which makes it less suitable for sharp and highly dynamic systems [7]. Its performance is very high on the accuracy of the system model, and impurities can cause sub- or unstable behaviour. Consequently, both PID and MPC struggle to meet modern robotic ARM control demands, where uncertainty, variable payload and human-robot cooperation are required for adaptability, strength and multi-purpose adaptation for the functions associated with the cooperation [8].

As mechatronic systems, including robot manipulators, become complex, robust and adaptive control strategies are needed that can accurately address nonlinear dynamics and external disturbances as well as parameter uncertainties in real time [9]. The familiar control techniques, such as PID controllers, are simple to implement, which is the reason why they are popular, but they have shown on many occasions to be unable to deliver satisfactory performance in such settings since they lack the natural flexibility and are unable to cope with high levels of nonlinearity. Recent innovation in artificial intelligence and bio-inspired algorithms promises to offer the solution to improve control systems [10]. In particular, the Echo State Networks (ESNs) models of reservoir computing offer an effective technique for reasoning about the time dynamics and system behaviour learning but with fewer computational demands. More so, bio-inspired techniques have demonstrated very good promise in optimising the parameters of complex systems and so are incredibly applicable in adaptive control [11].

In spite of these developments, there exists a gaping hole in the literature. The current state of the art in neural network controllers uses gradient-based optimisation whose representation and optimisation techniques have drawbacks associated with local minima and are not likely to capture multi-objective trade-offs between, e.g., tracking accuracy, energy consumption and stability. In addition, although evolutionary algorithms, such as Harris Hawks Optimisation (HHO) and Reptile Search Algorithm (RSA), have been shown to be successful in terms of optimisation at a single level, researchers have not examined their multi-level hybridisation to optimise the controllers of mechatronic systems based on deep learning. To resolve these challenges, the study proposes an adaptive control strategy based on ESN, in which hyperparameters are adapted using a hybrid HHO-RSA framework. The purpose is to develop a strong, energy-skilled and stable controller for gate 2-DOFrl robotic arm, which is capable of achieving better trajectory tracking in nominal and disturbed conditions. The approach aims to combine the rapid convergence and exploration capabilities of HHO with the local refinement strength of the RSA, which can control the boundaries of the existing control functioning. The key contribution of the research are as follows:

- A novel hybrid optimization algorithm combining Harris Hawks Optimization (HHO) and Reptile Search Algorithm (RSA) is developed to tune Echo State Network (ESN) parameters effectively.
- An adaptive ESN-based control system is designed for a 2-DOF robotic arm to handle nonlinear dynamics and external disturbances in real time.
- The proposed controller improves performance, achieving higher tracking accuracy, reduced energy consumption, and better stability compared to PID and LSTM controllers.
- Extensive simulations validate the approach, showing its effectiveness under both normal and disturbed conditions.

Rest of Section

- **Section 2:** Reviews recent adaptive control methods and bio-inspired optimization techniques for mechatronic systems, highlighting limitations of PID and traditional neural networks in handling nonlinear dynamics and external disturbances.
- **Section 3:** Details the proposed ESN-RSA-HHO framework, including the robotic arm dynamic modeling, ESN architecture, and hybrid optimization process for tuning controller parameters.
- **Section 4:** Presents experimental results and comparative analysis, showcasing tracking accuracy, energy efficiency, and stability improvements over baseline controllers.
- **Section 5:** Concludes the study by emphasizing the effectiveness and robustness of the proposed approach, and discusses future directions such as real-world hardware implementation and scalability to higher-DOF systems.

II. LITERATURE REVIEW

Pan et al. [12] paper introduces a bio-inspired composite learning control strategy to enhance tracking precision in robotic manipulators through compensation of frictional uncertainties. Drawing inspiration from cerebellar learning mechanisms, the technique combines proportional feedback with a memory-based error model, allowing for accurate control without demanding high feedback gains. The technique was implemented on a DENSO industrial robot arm and met enhanced transient and steady-state performance. This method minimizes the use of energy and guarantees robustness without sophisticated model dependence. The research proves that biologically inspired control systems are capable of outperforming conventional PID methods in real-world robotics, paving the way for adaptive learning for real-time manipulation operations.

Hu et al.[13] research combines Soft Actor-Critic (SAC), Long Short-Term Memory (LSTM), and Generative Adversarial Imitation Learning (GAIL) to develop a deep reinforcement learning framework for robotic trajectory control. The proposed model overcomes disturbances and complicated dynamics of robotic arms through learning from demonstrations of experts and real-time adaptation. Temporal dependencies are captured by the LSTM network, and GAIL optimizes the policy by imitation learning. The hybrid model is better in tracking precision, robustness, and constraint following compared to individual RL algorithms according to experiments. The method proves the efficacy of merging memory-based neural models with policy learning in adaptive control tasks in dynamic environments.

Zhu et al. [14] introduce a control system for a bionic quadruped robot employing a Multi-Objective Whale Optimization Algorithm (MOWOA) to improve gait adaptability and energy efficiency. The robot has a biologically inspired parallel torso and the control approach exploits model predictive control (MPC) optimized through MOWOA. This permits simultaneous optimization of multiple control goals like stability, speed, and energy efficiency. The paper proves enhanced motion coordination and environmental adaptability over diverse terrain conditions. This work highlights the potential of bio-inspired multi-objective optimization in improving the locomotion performance of advanced mechatronic systems like quadruped robots [14].

Boddhu et al. [15] presents evolutionary neurocontrollers for flapping-wing unmanned aerial vehicles (UAVs). The neural network parameters are optimized on aerodynamic stability, energy efficiency, and control smoothness using a multi-objective evolutionary algorithm. The work simulates dynamics of flapping flight and the performance of controllers in a variety of environmental conditions. Findings show that evolutionary deep learning methods facilitate adaptive behavior and fault tolerance and can thus be applied to highly nonlinear systems. While specifically discussing UAVs, the ideas are applicable to land-based mechatronic systems, justifying your proposal's application of sophisticated neural networks in adaptive control for robot arms or manipulators.

Mompó Alepuz et al. [2] discusses progress in brain-inspired control systems, with special attention to biomimetic robotics. It classifies adaptive and hybrid control approaches in accordance with biological mechanisms like central pattern generators, cerebellar learning, and cortical modulation. The article details novel methods that merge neural networks and optimization approaches for real-time learning and control in uncertain conditions. It further analyzes the trade-offs between engineering feasibility and biologically plausible control. Although it does not introduce new implementations, it offers theoretical grounding and background for applying bio-inspired

neural structures such as Echo State Networks (ESN) and evolutionary algorithms to mechatronic systems, mirroring your work.

Motoaki Hiraga et al. [16] examine the application of Echo State Networks (ESNs) for decentralized control of swarms of robots. According to their research, ESNs can learn behaviors through local learning and feedback in distributed multi-agent systems. Through the use of the reservoir computing function of ESNs, robots in the swarm can achieve adaptive, coordinated behavior with little time for training. The approach is bio-inspired, mimicking how simple neural circuits in animals produce collective behavior. Although applied to swarm robotics, the techniques are equally valuable for real-time adaptive control in single-robot mechatronic systems, especially in resource-constrained embedded environments.

Y. Li et al. [17] proposes a novel fuzzy Echo State Network (ESN) model with online learning capabilities designed to control redundant robotic manipulators. Motivated by cerebellar learning, the system progressively adjusts its weights on the basis of online feedback, and accuracy is enhanced with complex movements. The fuzzy layer makes the network more interpretable and also addresses nonlinearities well. Simulation and hardware results indicate that the model can easily adapt to varying tasks and environments with low computational expense. This paper is particularly germane to your study since it applies bio-inspired adaptive neural control directly to multi-degree-of-freedom robotic systems with online optimization efficiency.

Tham et al. [18] investigate how Echo State Networks (ESNs) might be made to mimic the action of biological Central Pattern Generators (CPGs), the agents behind animals' rhythmic movements. The work shows that a well-tuned ESN can produce stable oscillatory outputs required for walking or joint actuation without rhythmic inputs. The bio-inspired method makes the control system design of periodic motor robotic platforms less complex. The research offers foundational knowledge about employing ESNs as bio-inspired motor controllers and supports the promise of reservoir computing in adaptive as well as rhythmic control activities such as walking or grasping.

Banderchuk [19] paper introduces a hybrid control method that integrates classical robust control theory and learning with ESN to cope with uncertainties and sustained disturbances in nonlinear systems. The framework consists of a baseline robust controller and an ESN module with learning and compensation for modeling errors in real-time. The research illustrates that this hybrid strategy enhances stability and minimizes tracking error even in cases with uncertain external conditions. It is especially appropriate for robotic manipulators in dynamic settings. The combination of model-based and data-driven methods underpins the concept of adaptive ESN control with evolutionary tuning, such as in your intended work.

Galván et al. [20] examine the use of evolutionary multi-objective optimization (EMO) methods—more specifically NSGA-II and MOEA/D—to tune deep learning models to trace-prediction tasks. They optimize trade-offs among accuracy, model complexity, and energy usage using CNN and LSTM architectures. The research discovers that evolutionary optimization yields more generalizable and energy-efficient models than standard training methods. While applying to trajectory prediction specifically, the techniques are applicable to training any deep neural controller with competing objectives. This work aligns with your objective of optimizing ESN parameters via hybrid multi-objective approaches such as HHO–RSA for robotic arm real-time control performance.

W. K. [illegible] et al. [21] article introduces a multi-objective evolutionary design approach for Central Pattern Generator (CPG) networks applied to biomimetic robotic fish. The authors optimize CPG parameters with Pareto-based algorithms in order to trade-off between goals such as swimming efficiency, stability, and manoeuvrability. The method replicates biological spinal control mechanisms and confirms its viability through simulation. While concentrating on aquatic movement, the paper demonstrates how bio-inspired oscillatory control structures can be optimized with evolutionary algorithms. This reinforces your combination of ESN-based rhythmic control for joint actuation and hybrid optimization techniques for multi-goal dynamic mechatronic systems

Basterrech and Rubino [22]introduce an evolutionary approach to training Echo State Networks (ESNs) for time series prediction. Instead of using fixed reservoir parameters or gradient-based training, they evolve the network weights using metaheuristic algorithms to minimize inaccuracy and maximize robustness. It is computationally intensive but effective for nonlinear time-series data sets and demonstrates its potential in modeling dynamic systems. This research is in direct relevance to your own work because it supports applying hybrid bio-inspired optimization methods such as HHO–RSA to evolve parameters of ESN for adaptive control in robotic or mechatronic systems under real-time constraints.

III. PROPOSED FRAMEWORK ON BIO-INSPIRED ADAPTIVE CONTROL MECHANISMS IN MECHATRONIC SYSTEMS

This research, the proposed Echo State Network (ESN) in the system setup, includes a 2-DOF robotic arm configured to evaluate the adaptive control mechanism, adapted to the Hybrid Harris Hawks–Reptile Search Algorithm (HHO-RSA). The robotic arm consists of two rigid links of 0.3 m and 0.2 m with a respective mass of 1.2 kg and 0.8 kg, which are connected through the active rotary joints activated by high-torque DC servo motors, which are united for an accurate position and velocity response with 1024 PPR quadrature [23]. The system consists of current sensors (INA219) for the real-time monitoring of energy consumption and torque estimation, and an MPU9250 IMU has been occupied for end-influentially mounted dynamic states such as acceleration and adaptive control under orientation. The power supply is made through a regulated 24V DC source, in which the current handling is up to 5A to run both actuators firmly. The ESN controller is deployed within a Matlab/Simulink environment, which interfaces with the Mechatronic system model to generate torque commands, while the HHO-RSA optimiser tunes the reservoir parameters, spectral radius and output weight in an offline phase before real-time control execution. This setup enables verification of high-resolution sensor data collection and adaptive control strategy under various trajectory and disturbance landscapes, which ensures a strong evaluation of the proposed approach. Fig1 show present the proposed frame on bio-inspired adaptive control mechanisms in mechatronic systems:

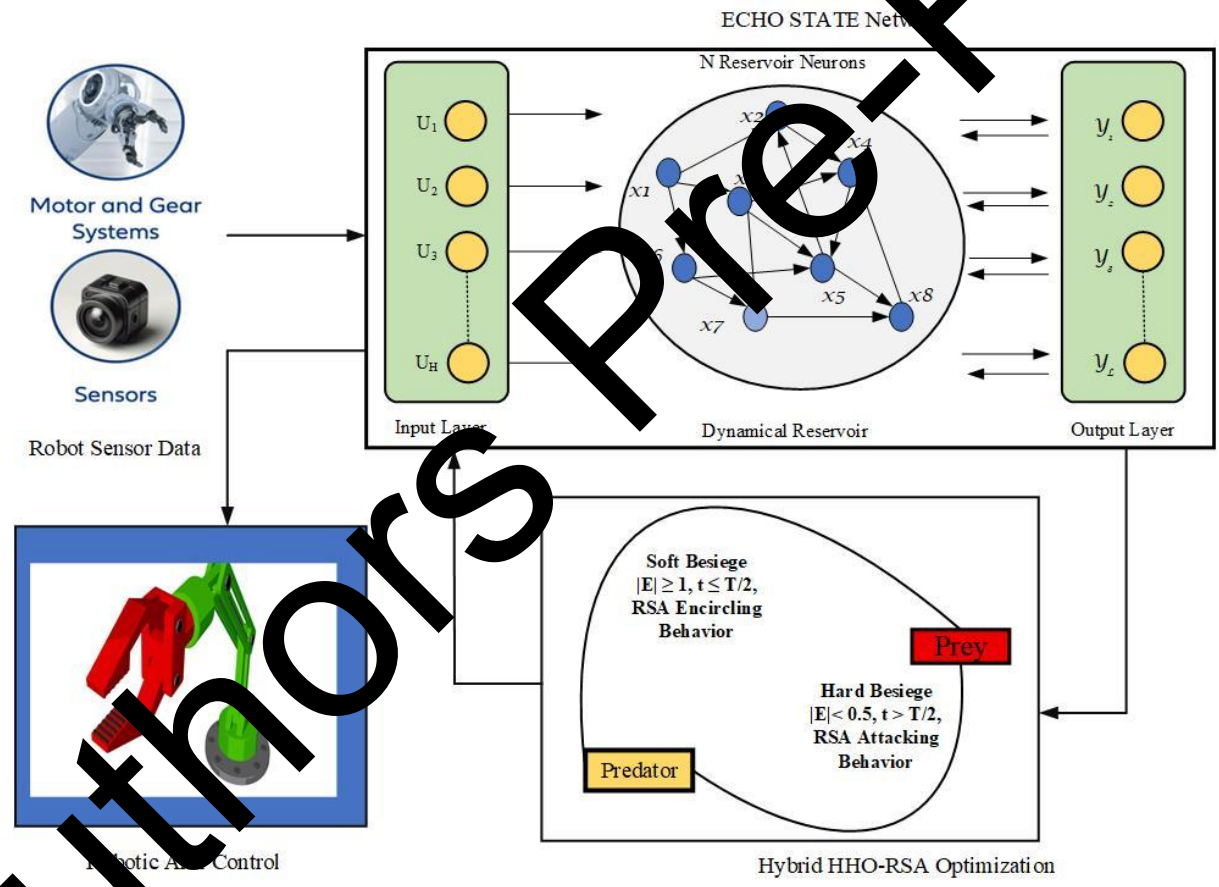


Fig 1. Proposed Framework on Bio-Inspired Adaptive Control Mechanisms in Mechatronic Systems

3.1 Sensor Data Acquisition and Variables

In the proposed system, sensor data acquisition is designed to capture the required variables required for training and evaluation of adapted ESN-based adaptive control using a hybrid HHO-RSA approach. The robotic arm is equipped with high-resolution rotary encoders (1024 PPR) on each joint to measure accurate angular positions and velocities, an MPU9250 Inertial Measurement Unit (IMU) to climb the final influence to catch the acceleration and orientation, and the INA219 current sensors unite [24]. The data collection process records the desired joint torque and target positions in the record input signal, while output reactions include real joint positions, velocity,

control errors and real-time energy use. To ensure high-loyal data for controller training, the system operates with a sample rate of 500 Hz, which captures enough temporary resolution to model dynamic behaviours accurately. Multiple trajectory profiles – steps, sinusoidal, and random reference inputs – are applied to the system to expose a variety of speed patterns and disturbances, causing the dataset to enrich the dataset and enable strong training and evaluation of the adaptive control strategy. The following Table 1 shows the sensor data requirements for adaptive control system:

Table 1: Sensor Data Requirements for Adaptive Control System

Sensor Type	Unit	Sampling Rate
Rotary Encoder	radians (°)	500 Hz
Rotary Encoder (derived)	rad/s	500 Hz
IMU (MPU9250)	m/s², degrees	500 Hz
Current Sensor (INA219)	Amperes (A)	500 Hz
System Input Logger	Nm or PWM (%)	500 Hz
Voltage Sensor (optional)	Volts (V)	100 Hz

3.2 Data Acquisition and Pre-processing

In the proposed adaptive control structure, data is acquired on the 2-DOF robotic ARM setup using a combination of high-resolution rotary encoders, an MPU9250 IMU, and INA219 current sensors to capture the dynamic reaction of the system under various trajectory profiles. Input signals in the dataset, such as the desired joint torques $\tau_d(t)$ and target positions $\theta_d(t)$, as well as actual joint angles $\theta(t)$, velocities $\dot{\theta}(t)$, control errors $e(t) = \theta_d(t) - \theta(t)$, and energy consumption E(t). To ensure the quality of data, the signals are sampled at 500 Hz and initially processed to reduce the sensor noise. The Butterworth low-pass filter of another order is applied to eliminate high-existing components, mathematically computed using Eqn (1):

$$y(t) = \frac{b_0x(t)+b_1x(t-1)+b_2x(t-2)-a_1y(t-2)-a_2y(t-2)}{a_0} \quad (1)$$

Where the raw signal and the filtered output is denoted as x (t) and y (t).

After filtering, all variables are normalised to the limit [0, 1]. Using min-max scaling to improve the stability and convergence of Echo State Network (ESN) during training and computed using Eqn (2):

$$x_{norm} = \frac{x-x_{mn}}{x_{max}-x_{min}} \quad (2)$$

Where the minimum and maximum observed value in the dataset is denoted as $x_{max}\ and\ x_{min}$.

To prepare time-series data for the ESN reservoir, generalised signals are divided into overlapping time windows creating input-output pairs (U_i, Y_i) to training. Each input sequence is calculated using Eqn (3) – (4):

$$U_i = [u(t_i), u(t_i + 1), \ldots., u(t_i + w - 1)] \quad (3)$$

$$Y_i = [y(t_i), y(t_i + 1), \ldots., y(t_i + w - 1)] \quad (4)$$

Where u (t) control is input and y (t) is the related system reaction to the window. These preprocessed datasets, the hybrid Harris Hawks-Reptile Search Algorithm (HHO-RSA), enable the strong training and evaluation of the optimised ESN controller.

3.3 System Modelling and Echo State Network Integration

In this research, we developed an adaptive control structure for the 2-DOF robotic arm that operates in a planner scope. The system integrates a dynamic model of a robotic arm with an intelligent controller based on an Echo State Network (ESN), which is adapted to a combination of Harris Hawks Optimisation (HHO) and Reptiles Search Algorithm (RSA) using a hybrid evolutionary algorithm. The dynamic model of the robotic arm is obtained using the Lagrangian formulation, which captures its nonlinear characteristics, inertia, Coriolis effects and gravitational forces. This model provides a realistic representation of the manipulator's behaviour under separate input torque and disturbances [25]. ESN acts as a core controller in the system, using a certain recurrent reservoir network with the input signal to process input signals to include tracking errors, their derivatives and previous control actions. The reservoir maps these inputs into high-dimensional interior states, and only the output weight is adjusted to generate adaptive torque commands for robotic joints. To customise the performance of ESN, the Harris Hawks – Reptile Search (HHRS) hybrid optimiser is employed. This gives efficient tuning of significant hyper parameters, including spectral radius, input scaling, leakage rate and output weight of the reservoir, reduce tracking errors and ensure smooth energy-skilled operations. The entire control architecture operates in a closed loop, where the real-time sensor response from the joint encoder and current sensor is constantly fed to the ESN controller. This enables the robotic arm to follow the desired trajectory with high precision, adapting to dynamic uncertainties and external disturbances [26]. The simulations were made using MATlab/Simulink to validate the proposed approach under various trajectory profiles and states of disturbance. The 2-DOF robotic arm has been modernised as a planner manipulator, made up of two rigid links associated with rotary joints, which enables speed in a two-dimensional aircraft. The dynamic behaviour of this system is taken using Lagrangian formulation, which is an energy-based method for the installation of manipulator motion. Lagrangian L is defined as the difference between the total kinetic energy (T) and potential energy (V) of the system. The Lagrangian L is represented in Eqn (5):

$$L = T - V \tag{5}$$

Where the total kinetic energy of the system and the total potential energy is denoted as T and V.

In our system, the 2-DOF robotic arm contains two rigid links actively by rotary joints, each of which contributes to the overall kinetic energy of the mechanism. The kinetic energy link arises from both translational speed of mass and rotational motion about joints. Keeping in mind the speed in a planner scope, energy contribution is calculated relative to the centre of mass and rotational inertia of each link. This makes the base to capture the dynamic behaviour of the hand in speed, which is necessary for accurate control design. The kinetic energy T of the robotic arm is given by:

$$T = \frac{1}{2}m_1\dot{r}_1^2 + \frac{1}{2}I_1\dot{\theta}_1^2 + \frac{1}{2}m_2\dot{r}_2^2 + \frac{1}{2}I_2(\dot{\theta}_1 + \dot{\theta}_2) \tag{6}$$

In Eqn (6), the masses of link 1 and link 2 is denoted as $m_1\ and\ m_2$, the linear velocities of the centers of mass of link 1 and link 2 is denoted as $\dot{r}_1\ and\ \dot{r}_2$. The moments of inertia about the centers of mass for link 1 and link 2 is denoted as $I_1\ and\ I_2$ and the joint angular velocities is defined as $\dot{\theta}_1\ and\ \dot{\theta}_2$.

The robotic arm moves in the horizontal plane, and all the mass of links is affected by gravity and adds to the potential energy of the system. The overall potential energy depends on the position of the centres of mass of the links with respect to the fixed base frame. It is of great importance that this energy representation can be used to determine the influence of gravity on the motion of the manipulator, and the dynamic equation of the manipulator can be worked out to design the control. The potential energy V of the robotics arm is represented using the following eqn (7):

$$V = m_1 g l_{c1}\cos\theta_1 + m_1 g[l_1\cos\theta_1 + l_{c2}\cos(\theta_1 + \theta_2)] \tag{7}$$

In Eqn (7), the gravitational acceleration is denoted as g, the distanced from the joints to the center of mass between the link (1 and 2) is denoted as $l_{c1}\ and\ l_{c2}$.

The equations of speed for the robotic arm are obtained by applying the Euler-Lagrange equation to each joint:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \dot{\theta}_i} = \tau_i \tag{8}$$

The above Eqn (8) leads to the coupled non-linear dynamic equation as represented in Eqn (9):

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \tag{9}$$

In Eqn (9), the inertia matrix representing the system mass distribution is denoted as $M(\theta)$ and the coriolis and centrifugal matrix is denoted as $C(\theta, \dot{\theta})$ this helps in accounting for dynamic coupling and velocity-dependent effects. The gravity torque vector is denoted as $G(\theta)$ and finally the vectors control torques applies at the joints is denoted as $\tau = [\tau_1, \tau_2]^T$.

3.3.1 ESN-Based Adaptive Control

In the proposed control architecture, an Echo State Network (ESN) is used to generate adaptive torque commands for a 2-DOF robotic arm. The ESN reservoir is a type of recurrent nervous network based on computing paradigm, which is well suited for temporal dependence and the modelling of dynamic systems. The ability to capture non-linear characteristics of a robotic arm makes it ideal for adaptive control in an uncertain environment. The ESN architecture consists of three main components: input layer, reservoir (dynamic core), and output layer.. Fig2 shows the architecture of ESN Ratting:

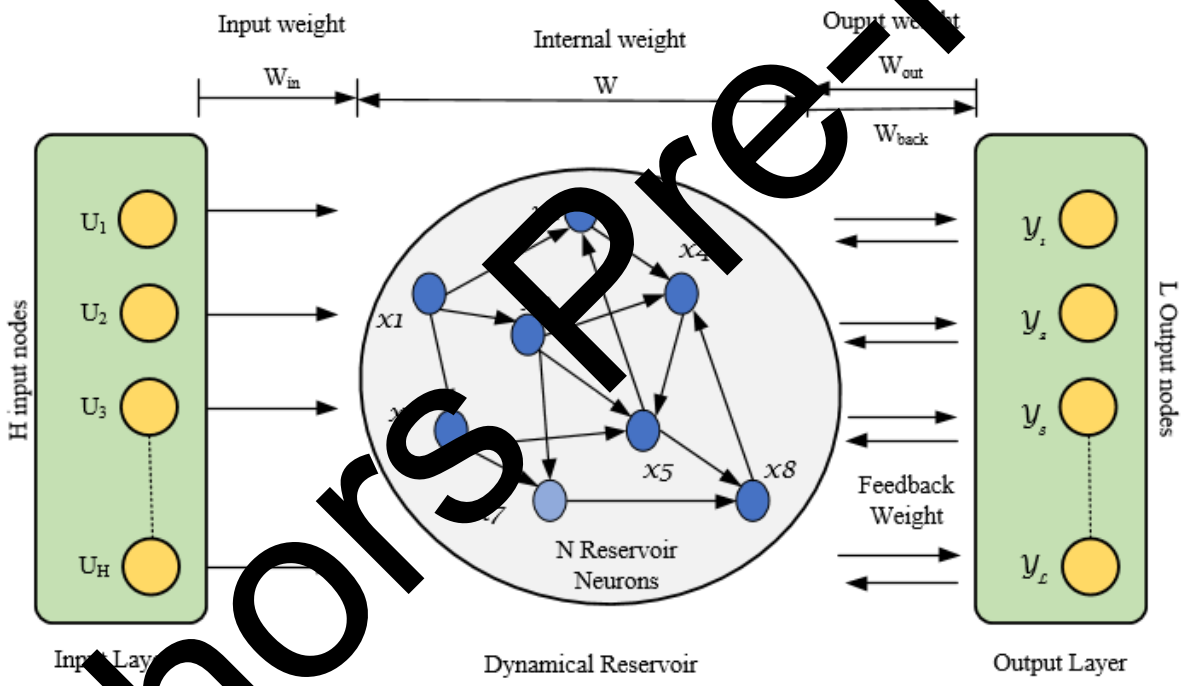


Fig2: Echo State Network Ratting

Input Layer

The input layer encounters the current position of the system and controls the objectives in a feature vector that feeds into the reservoir. The primary input includes tracking error $e(t)$, which is defined as the desired joint positions $\theta_d(t)$ and real joint posts $\theta(t)$, error $\dot{e}(t)$, and the difference from the previous control input. These allow input ESN to learn relations between system errors and corrective actions required for precise trajectory tracking.

Reservoir (Dynamic Core)

ESN has a large, very associated recurrent nerve network in the heart called reservoirs, characterised by a certain internal weight matrix W_{res}. The reservoir converts low-dimensional input into a high-dimensional dynamic state location, providing temporary mobility to learning. The reservoir state $x(t)$ develops according to the following updated rules:

$$x(t) = (1 - \alpha)x(t-1) + \alpha.\tanh(W_{in}u(t) + W_{res}x(t-1)) \quad (10)$$

In Eqn (10), the reservoir state vector at time t is denoted as $x(t)$ and the input vector containing the tracking error signals and past control data is denoted as $u(t)$. The input weight matrix that maps the input vector to the reservoir is denoted as W_{in}, the leakage rate controlling the speed of reservoir state transitions is denoted as α. This structure ensures that the reservoir captures both the short-term and long-term dependence required to control the non-linear robotic system.

Output Layer

The reservoir makes a linear readout of the reservoir state to generate control torque applied to output layer robotic joints:

$$\tau\,(t) = W_{out}x(t) \quad (11)$$

Here, the W_{out} shows the weight matrix, the only trainable component in the ESN. These weights are adapted to reduce trajectory tracking errors and ensure smooth control action. To train the ESN for strong performance, significant parameters – which include spectral radius, input scaling, leakage rate and output weight adapted Hybrid Harris Hawks–Reptile Search Algorithm (HHRS). This adaptive process enables ESN to customise non-linearity, uncertainties and external disturbances in the robotic system.

3.3.2 Hybrid RSA–HHO Optimization for ESN-Based Adaptive Control

In the proposed adaptive control strategy for the 2-DOF robotic arm, the Echo State Network (ESN) serves as a core controller to generate torque commands. The performance of ESN is highly dependent on its hyper parameters, including the spectral radius (ρ), input scaling (σ_{in}), leak rate (α), and output weight (W_{out}) Outside, To achieve the optimal tuning of these parameters, a hybrid optimisation framework is designed by combining the Reptile Search Algorithm (RSA) for local exploitation and Harris Hawks Optimization (HHO) for global exploration. This sequential hybridization ensures both fine parameter tuning and the ability to avoid local optima.

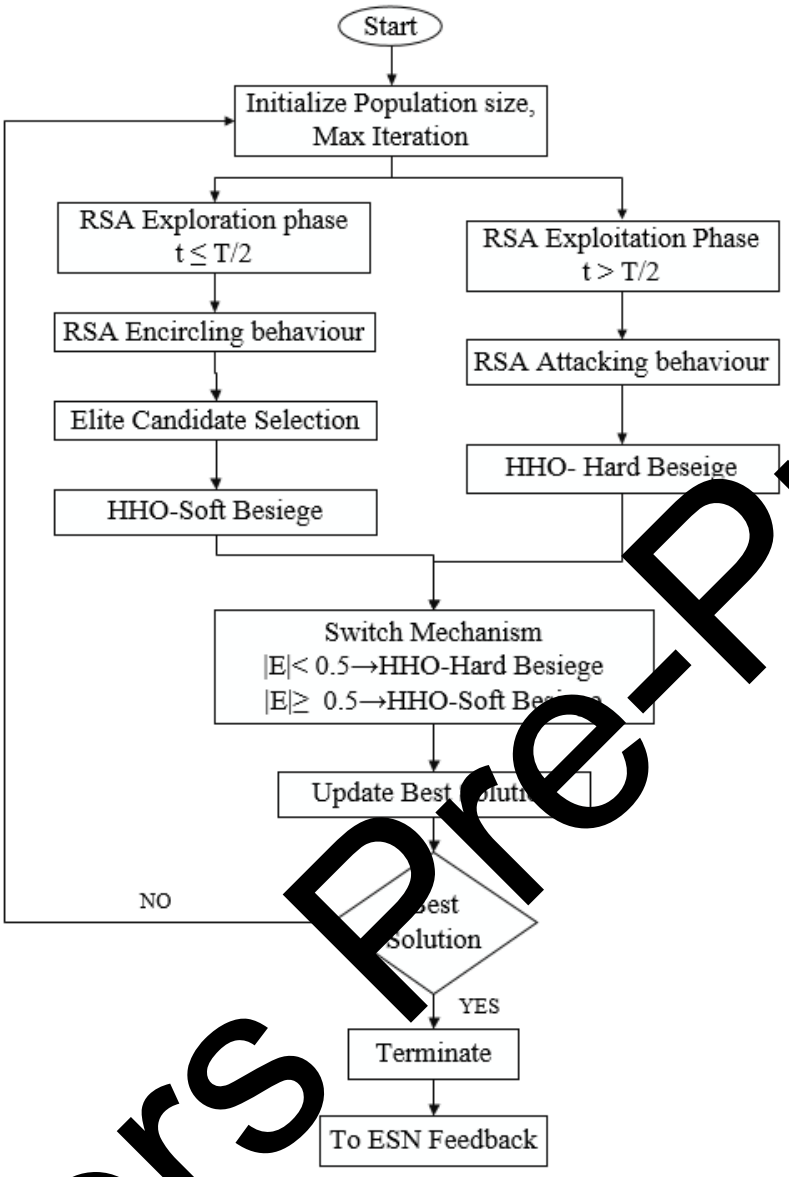


Fig 3. HHRS Optimization

Within the above mentioned hybridised frame of optimisation, one would first develop the Reptile Search Algorithm (RSA) that would initialize a batch size random parameterisation of Echo State Networks (ESN) and conduct an adaptive local search to optimise the corresponding ESNs within their immediate vicinity. In this step, the main hyper parameters can be optimised, including spectral radius, input scale, leakage gap, and output weights to allow the ESN to attain better local performance. After RSA has finished its exploitation phase, the optimised parameter sets are transferred to the stage of the Harris Hawks Optimisation (HHO), which uses the population of RSA optimised solutions as initial solutions and engages in a global search across the expanded parameter space. HHO adopts exploration and exploitation solutions based on collaborative foraging of Harris hawks to evade local minima and become close to globally ideal solutions. The above Fig3 shows the HHRS optimization:

Stage 1: Reptile Search Algorithm

The adaptation process begins with RSA, which is inspired by the adaptive hunting strategies of reptiles such as crocodiles. The RSA candidate focuses on local exploitation by making proper adjustments to the ESN parameter set.

Initialization

When each x_i represents a vector of ESN parameter then population of candidate solution $X = [x_1, x_2, \ldots . x_n]$ is generated as represented in Eqn (12):

$$x_i = [\rho_i, \sigma_{ini}, \alpha_i, W_{outi}] \tag{12}$$

Within the predefined bounds the parameters are initialized.

Local Exploitation

Using the adaptive position update rule each candidate x_i search its neighbourhood as represented in Eqn (13

$$x_i^{t+1} = x_i^t + S.rand.(x_{best}^t - x_i^t) \tag{13}$$

Where the current position of candidate i at iteration t is denoted as x_i^t, the best solution so far and it is denoted as x_{best}^t. The adaptive step size and the random number in [0,1] is denoted as S and ra

Intermediate Solution Set:

RSA provides a refined set of locally optimized ESN parameter after a fixed number of iteration.

Stage 2: Harris Hawks Optimization (HHO)

The intermediate solution from the RSA is then passed to the HHO fo global exploration. HHO is inspired by the cooperative hunting behaviour of Harris hawks and uses strategies such as "soft bases" and "hard bases" for convergence on global optima.

Exploration Phase

Using the update rule the Hawks randomly perch aro d the search space as represented in Eqn (14):

$$x_i^{t+1} = x_i^t + rand.|x_i^t - 2.rand.x_{rand}^t| \tag{14}$$

In Eqn (14) the randomly selected and e is denoted as x_{rand}^t.

Transition Mechanism

Based on Eqn (15) the escape ener y E of the prey is computed:

$$E = 2\left(1 - \frac{t}{T}\right) \tag{15}$$

Where the current recurrence and the number of recurrences in the maximum number. It controls whether Hawk emphasises exploration ($|E| \geq 1$) or exploitation ($|E| < 1$).

Exploitation Phase

During the exploitation mode, the hawks surround the prey and update their positions as represented in Eqn (1

$$x_i^{t+1} = x_{prey}^t - E.\left|J.x_{prey}^t - x_i^t\right| \tag{16}$$

Where the random jump strength is denoted as J.

Multi-Objective Fitness Function

Both RSA and HHO use a multi-purpose fitness function J Each candidate is to evaluate the ESN parameter set as mentioned in Eqn (17):

$$J = \alpha_1 . MSE_{track} + \alpha_2 \, Energy_{consumed} + \alpha_3 (1 - Stability_{acore}) \quad (17)$$

Where MSE_{track} denote the Mean squared error between the desired and actual trajectories, $Energy_{consumed}$ denoted as total amount of energy used by robotic arm. $Stability_{acore}$ denote the qualification of the systems stability (e.g. according to Lyapunov) within a metric. $\alpha_1, \alpha_2, \alpha_3$ denote the Weighting parameters of accuracy, energy and stability.

During either stage, a multi-objective fitness function is used to measure the performance of every candidate solution and strike a balance between tracking accuracy, energy consumption, and stability. The proposed adaptive bandwidth integration continues through a process of repetition until iteration convergence conditions are satisfied and the optimal ESN parameters are chosen to track the trajectories of the robotic arm. The given method of RSA-first and HHO-second hybridisation guarantees that the initial stages of the process, during which parameters are fine-tuned accurately, and subsequent stages when the global optimization of the process is provided, making the tracking process both more accurate and energy-efficient, as well as highly controllable. In the closed-loop control system, the ESN sensor processes feedback and produces adaptive torque. In real time $\tau(t)$, which are also applied to the robotic arm to achieve smooth and accurate speed under disturbances and modelling uncertainties.

IV. RESULT AND DISCUSSION

The development and simulation of a 2-DOF robotic arm model with more realistic mechanical and dynamic parameters were also performed to test the performance of the proposed adaptive control strategy in this study. It involved an Echo State Network (ESN) optimally designed controller that was connected to a robotic system utilising the Hybrid Harris Hawks Reptile Search Algorithm (HHO-RSA). Within a closed-loop format, the ESN-based controller was used to produce the torque instructions that allowed fine trajectory following in nonlinear dynamics and the presence of external interferences. The properties of the mechanical structure, dynamics, and control system have been verified by MATLAB/Simulink simulation and formed a strong basis to test the accuracy of tracking, energy consumption, and stability of the system.

4.1 Robotic Arm Design Parameters

In this study, a 2-DOF robotic arm was designed with realistic mechanical specifications, including a link length of 0.3 m and 0.2 m, a payload capacity of up to 1 kg, and servo motors equipped with encoders for accurate joint control as mentioned in Table 2. The system integrates the sensor response from the encoder, an IMU and a torque sensor to monitor the position, velocity and applied forces within the ± 180° joint limit. This configuration provides a strong platform to implement and evaluate the customised ESN-based adaptive control strategy using the hybrid HHO-RSA algorithm.

Table 2: Mechanical Parameters

Parameter	Value (Typical)	Description
Degrees of Freedom	2 or 3	Shoulder + elbow + optional wrist
Link 1 Length	0.3 m	First arm segment (shoulder to elbow)
Link 2 Length	0.2 m	Second segment (elbow to wrist)
Link 3 Length	0.1–0.15 m	(Optional) Wrist link
Payload Capacity	0.5–1.0 kg	Max object mass at end-effector
Total Mass	2–3 kg	Structure + motors
Motor Type	Servo/DC w/ encoders	Closed-loop control
Sensor Feedback	Encoders, IMU, Torque	For joint position, velocity, and force
Joint Range	±180° (all joints)	Full rotational range (in radians: $\pm\pi$)
Drive	Direct or geared	Based on torque requirements

Table 3: Dynamic Parameters

Symbol	Description	Typical Values
m_1, m_2	Link masses (kg)	1.2 kg, 0.8 kg
l_1, l_2	Link lengths (m)	0.3 m, 0.2 m
I_1, I_2	Link inertias (kg·m²)	0.01, 0.008
τ_1, τ_2	Input torques	From controller
θ_1, θ_2	Joint angles	In radians
G	Gravitational acceleration	9.81 m/s²
j_1, j_2	Joint damping coefficients	0.01–0.05

Table 3 dynamic parameters of the robotic arm include link mass (1.2 kg and 0.8 kg), link length (0.3 m and 0.2 m), and joint inertia, which defines its speed characteristics. Gravity acceleration (9.81 m/s) and the combined soaking coefficient (0.01–0.05) models are included in realistic mobility. These parameters are important for accurate torque calculation and simulation in ESN-based adaptive control structures.

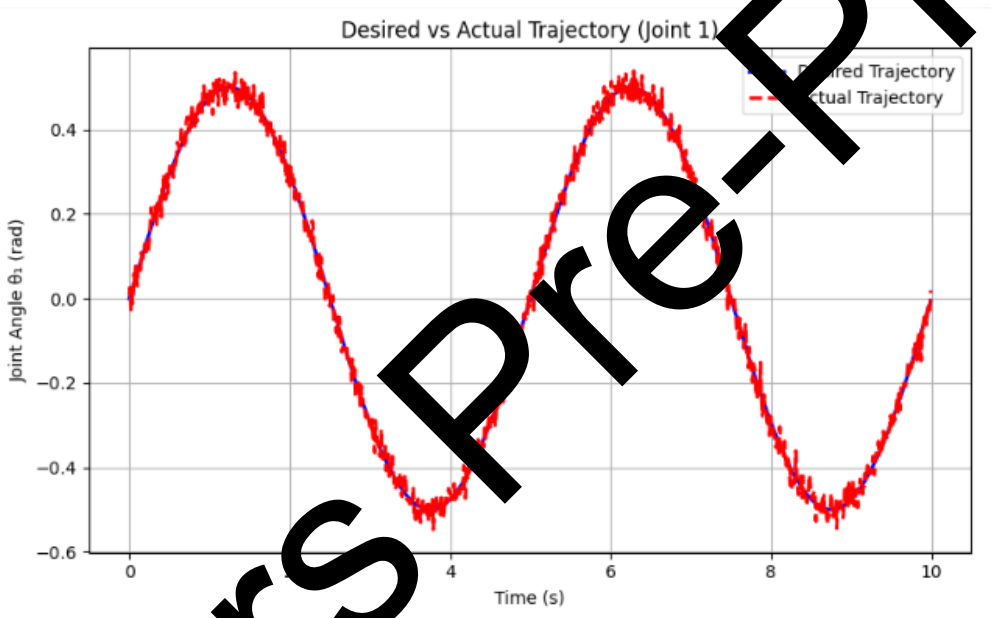


Fig Desired vs Actual Trajectory (Joint 1)

Fig 4 shows a comparison between desired and Actual joint trajectory at intervals of 10 seconds for joint 1. The desired trajectory, which is represented by a smooth blue sinusoidal curve with the dimension of 0.5 radians, defines the ideal motion profile for the joint, with a frequency of 0.2 Hz. In contrast, the actual trajectory, shown as a red dashed line, follows the desired path closely but displays minor deviations due to the presence of small random noise, simulating tracking errors contained in real-world control systems. The plot highlights the system's ability to maintain accurate tracking performance, capturing the slight changes between the desired and obtained joint angles.

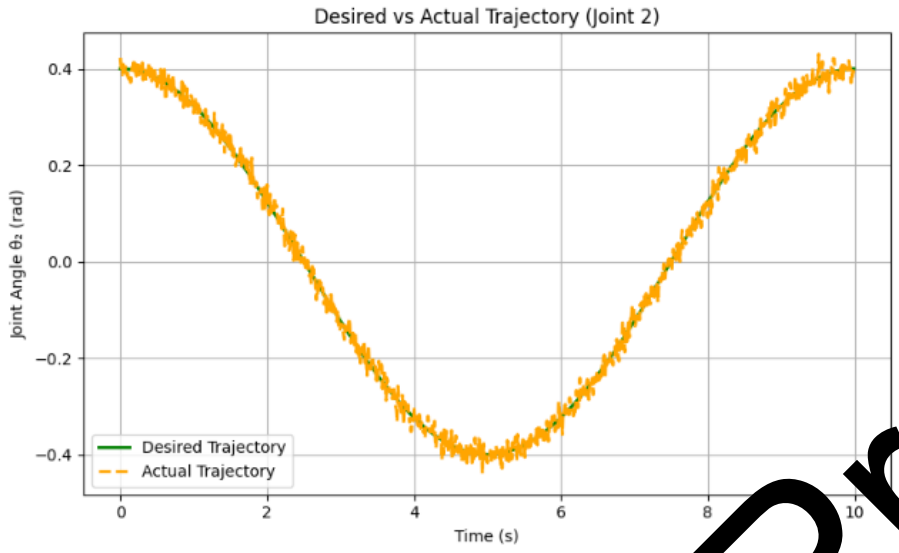


Fig 5. Desired vs Actual Trajectory (Joint 2)

Fig 5 presents a comparison between the desired and actual combined trajectory for joint 2 in the 10-second period. The desired trajectory, which is represented as a smooth green cosine wave with the dimensions of 0.4 radius and the frequency of 0.1 Hz, represents the target speed profile for the joint. The actual trajectory shown by the orange collapse line tracks the desired path closely but involves minor ups and downs due to small random noise, simulating trekking errors usually faced in practical control systems. The view shows the ability of the system to achieve accurate trajectory tracking effectively with minimal deviation between desired and real combined angles.

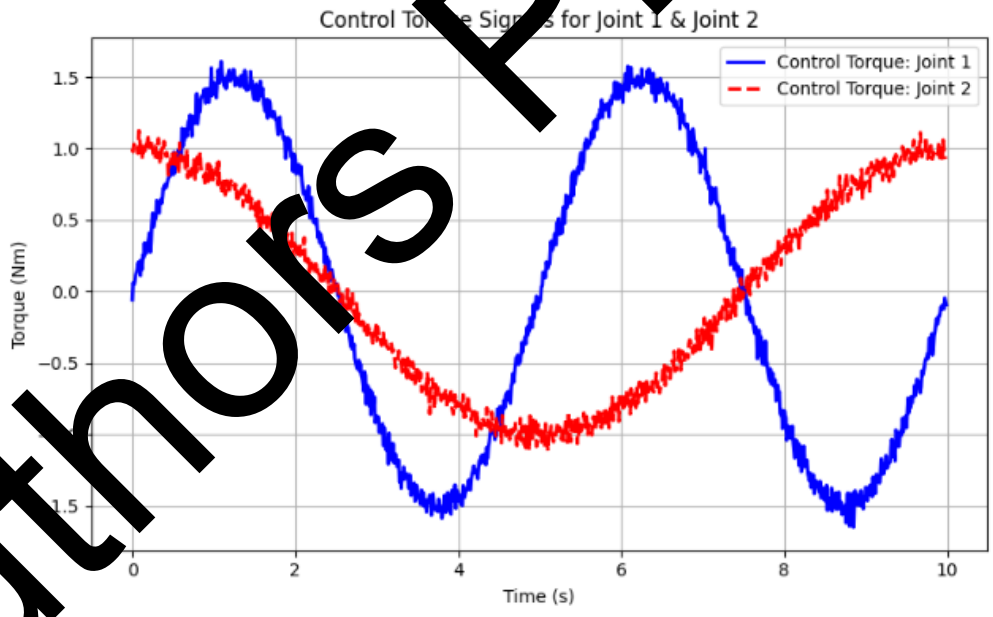


Fig 6. Control Torque Signals for Joint 1 and 2

Fig 6 control trorque 10 seconds shows the control torque signals applied to joint 1 and joint 2. The blue curve represents the joint 1, which shows a sinusoidal torque profile with dimensions of 1.5 nm and a minor random variety, which simulates the noise. The red collapse curve corresponds to the combined 2, characterising a cosine torque pattern with a low dimension of 1.0 Nm and similar noise effects. This visualisation highlights the dynamic torque input required to obtain accurate trajectory tracking in both joints under realistic conditions.

Table 4: Stability and Robustness Analysis

Test Condition	Tracking RMSE (rad)	Overshoot (%)	Settling Time (s)	Stability Score
Nominal Trajectory	**0.012**	2.8	0.4	0.95
External Disturbance	**0.018**	3.5	0.5	0.92
Model Parameter Change	**0.016**	3.0	0.45	0.93

Table 4 shows the results of stability and robustness analysis of the control system under nominal trajectory, external disturbance and the change of a model parameter, and conditions. In the case of the nominal trajectory, the system would have a tracking RMSE of 0.012 radians, the minimum overshoot would be 2.8% and the settling time would be only 0.4 seconds to produce a high stability score of 0.95. Before external disturbance, the tracking error (s), overshoot, and settling time are improved by slight lapses (0.018 rad, 3.5%, 0.5 s respectively), and the stability mark is also slightly low (0.92). In the same way, when the model parameters are changed, the system remains stable with the tracking RMSE of 0.016 radians, 3.0 per cent overshoot, 0.45 s settlement time, and stability score of 0.93. These findings show that the system can meet stable and robust performance in changeable situations.

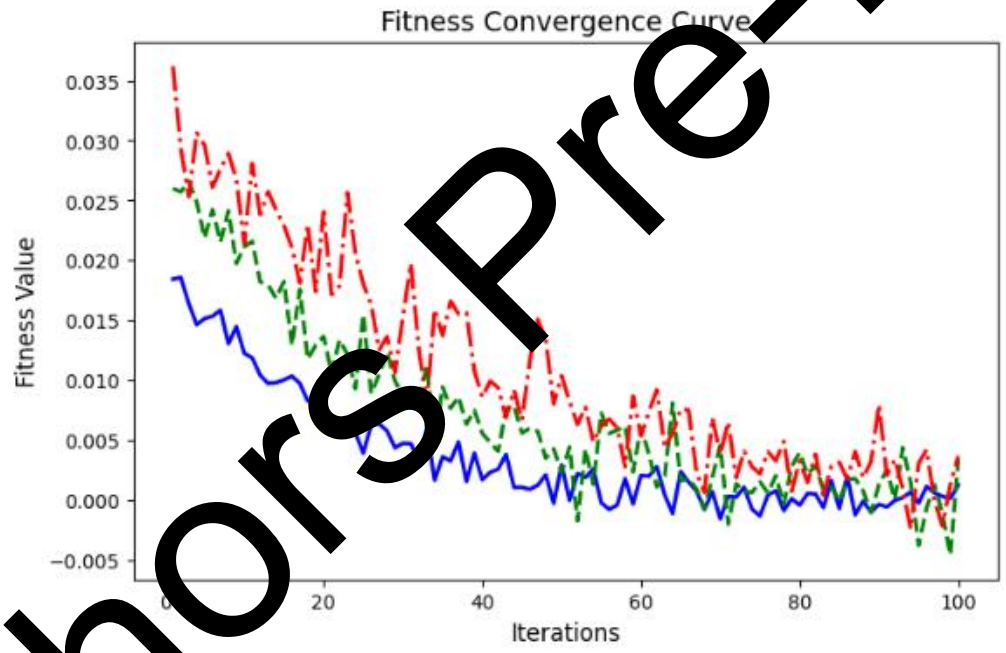


Fig 7. Fitness Convergence Curve

Fig 7 shows the fitness convergence curve of RSA-HHO (proposed hybrid), HHO, and RSA optimisation algorithms with three iterations of 100. The highest optimisation efficiency is depicted by the solid blue line or proposed RSA-HHO method, which converges to a low fitness value faster. It appears to contain a better optimization ability. By comparison, HHO only (green dashed line) and RSA only (red dash-dotted line) converge more slowly and settle at values of fitness that are higher. This is a representation of the demonstration of the improved performance of the RSA-HHO hybrid that tends to be faster and more effective at optimisation than its respective parts.

Fig 8. Parameter Evolution Over Optimization

Fig 8 compares tracking error over time for nominal and disturbing scenarios in a control system. In the nominal scenario shown by the solid blue line, the tracking error starts with about 0.05 radians and decreases rapidly, reaching zero due to the system being stable. In contrast, the distracted landscape depicted by the red collapse line, due to external disturbances, displays a higher initial error of about 0.08 radians with noticeable oscillations but gradually converts to minimal error. This highlights the strength of the visualisation system and the ability to effectively reduce trekking deviations under both general and challenging operating conditions.

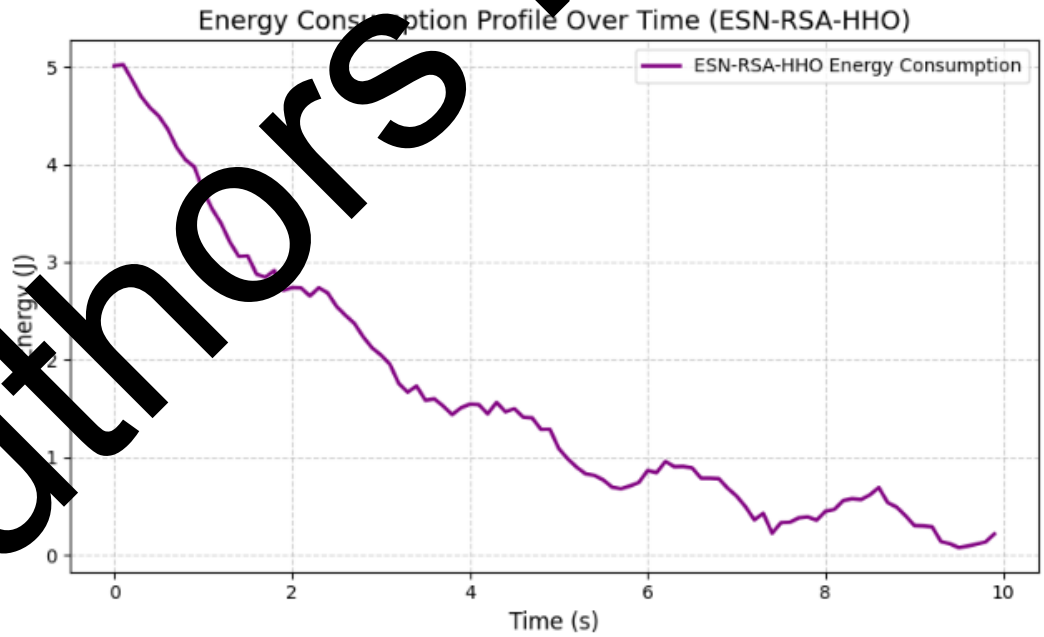


Fig 9. Energy Consumption Profile Over Time

Fig 9 shows the energy consumption profile of the ESN-RSA-HHO controller over a period of 10 seconds. Initially, energy consumption is relatively high but decreases rapidly as the system represents the convergence of the stabilisation process. Superimposed sinusoidal indicate transient adjustment in ups and downs control

attempts, while small random variations mimic practical noise effects. Overall, the curve displays the capacity of the controller to reduce the use of energy over time and receive a stable and customised energy profile.

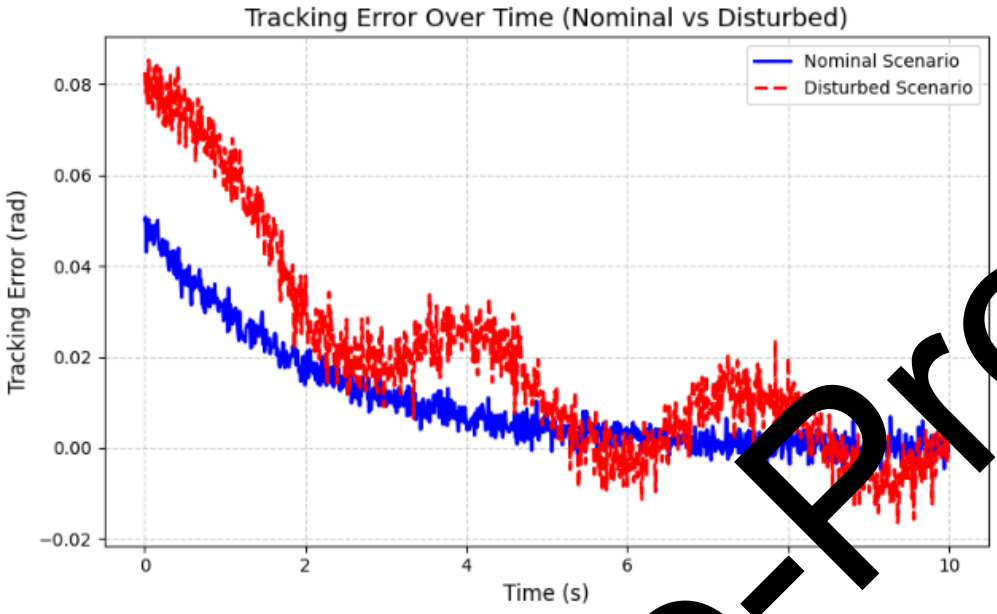


Fig 10. Tracking Error Over Time

Fig10 compares tracking error over time for nominal and disturbing scenarios in a control system. In the nominal scenario shown by the solid blue line, the tracking error starts with about 0.05 radians and decreases rapidly, reaching zero due to the system being stable. In contrast, the distracted landscape depicted by the red collapse line, due to external disturbances, displays a high initial error of about 0.08 radians with noticeable oscillations but gradually converts to minimal error. This highlights the strength of the visualisation system and the ability to effectively reduce trekking deviations under both general and challenging operating conditions.

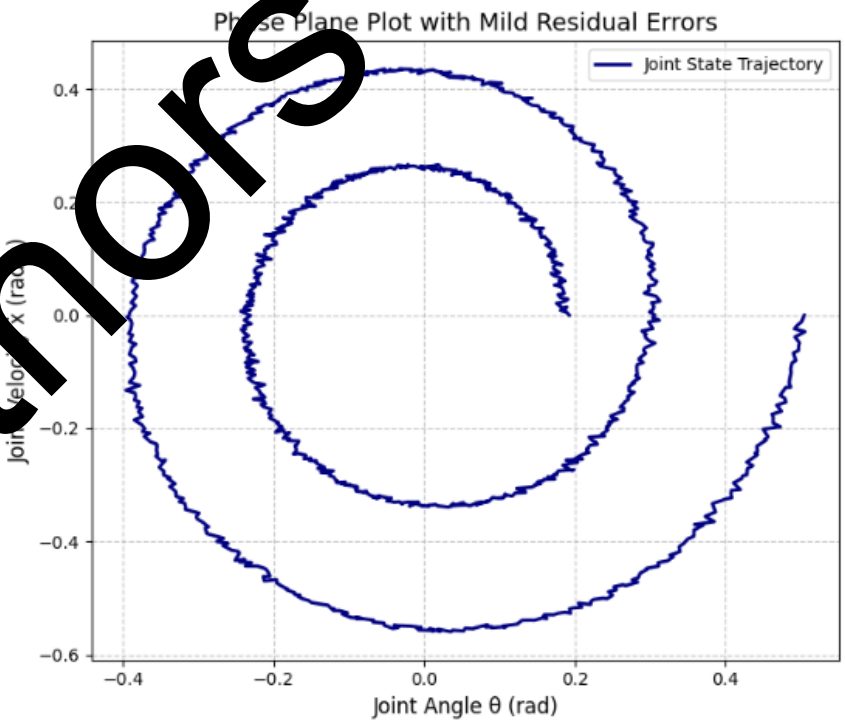


Fig 11. Joint State Trajectory

Fig11 presents a plane plot showing a plane aircraft depicting the United States trajectory with joint angles θ θ on the X-axis and joint velocity θ over time on the y-axis. The trajectory, depicted by the smooth naval curve, displays a spilling pattern that gradually converts towards the origin, showing the soaking behaviour and stabilisation of the system. Light residual errors are introduced as small random ups and downs, barely noticeable due to low noise, highlighting the effectiveness of the controller in reducing dynamic oscillations. This visualisation effectively demonstrates how the system infections reach a stable equilibrium point in the initial oscillator phase space.

Performance Metrics

Root Mean Square Error (RMSE)

RMSE measures the average quantity of tracking error between the desired trajectory and the real trajectory and computed using Eqn (18):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\theta_d(i) - \theta(i))^2} \quad (18)$$

Where the total number of samples were represented in N, the desired joint and the actual joint angle a time i is defined as $\theta_d(i)$ and $\theta(i)$. A lower RMSE indicates better trajectory tracking accuracy.

Total Energy Consumption

Total energy consumed by actuators during trajectory tracking.

$$E_{total} = \int_0^T \sum_{j=1}^{n} |\tau_j(t).\dot{\theta}_j(t)|dt \quad (19)$$

Where in Eqn (18), the control torque at joint j and the joint velocity at joint j is denoted as τ_j and $\dot{\theta}_j(t)$. The total simulation time and number of joint is denoted as T and n. The lower energy consumption reflects efficiency.

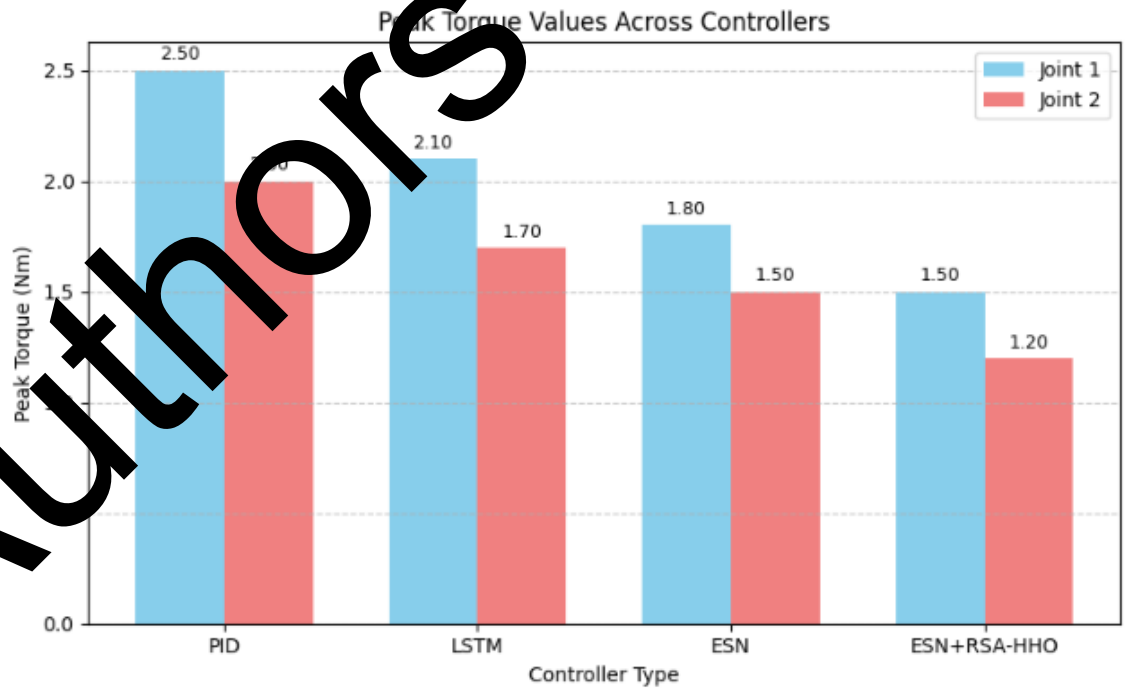


Fig 12. Peak Torque Across Controllers

Fig12 compares the peak torque values required by various controllers – PID, LSTM, ESN, and ESN+RSA-HHO – for joint 1 and joint 2. For joint 1, the peak torque is reduced from 2.5 nm to 1.5 nm with the PID controller, which uses ESN+RSA-HHO, indicating improper efficiency in torque. Similarly, for joint 2, peak torque is reduced from 2.0 Nm to PID for PID for PID for ESN+RSA-HHO. This visualisation highlights the better performance of the ESN+RSA-HS-HHO controller in reducing peak torque, demonstrating its effectiveness in achieving smoother and more energy-efficient joint controls than traditional and other advanced controllers.

Table 5: Comparative Tracking Performance

Controller Type	RMSE (rad)	Max Error (rad)	Rise Time (s)	Settling Time (s)	Overshoot (%)
PID Controller	0.035	0.110	0.4	0.8	7.5
LSTM Controller	0.021	0.065	0.35	0.6	5.2
ESN (no optimization)	0.018	0.055	0.32	0.5	4.0
ESN + RSA-HHO (Proposed)	0.012	0.042	0.28	0.4	2.8

Table 5 compared the tracking performance of four controllers – PID, LSTM, ESN (without optimisation), and using major metrics such as proposed ESN + RSA-HHO-RMSE, maximum error, increase time, settling time and overshoot. The PID controller shows the maximum error of the highest RMSE of 0.035 Radion and 0.110 Radion, with a time of growing of 0.4 seconds, a time of 0.8 seconds, and an overshoot of 7.5%. The LSTM controller improves performance with an RMSE of 0.021 Radion and lower overshoot at 5.2%. ESN (no adaptation) reduces further errors, receives 0.018 radians, 0.055 radius maximum error and RMSE of 4.0% overshoot. The proposed ESN + RSA-HHO displays the best performance in all metrics, 0.012 The lowest RMSE of Radians, 0.042; the maximum error of radio, fast growth of 0.28 seconds, the shortest time of 0.4 seconds, and the minimum overshoot of 2.8%.

Table 6: Energy Consumption and Control Effort

Controller Type	**Total Energy (J)**	**Avg. Control Torque (Nm)**	**Torque Variability (Std Dev)**
PID Controller	6.5	1.2	0.3
LSTM Controller	5.8	1.1	0.25
ESN + RSA-HHO (Proposed)	4.7	1.0	0.18

Table 6 presents a comparison of PID, LSTM and the proposed ESN + RSA-HHO in terms of energy consumption and control effort between controllers. The PID controller displays the highest total energy consumption at 6.5 J, with an average control torque of 1.2 Nm and torque variability of 0.3 Nm. The LSTM controller shows better efficiency, reducing energy consumption by 5.8 J, average torque to 1.1 Nm and torque variability by 0.25 Nm. The proposed ESN + RSA-HHO receives the best performance, with the lowest total energy consumption of 4.7 J, reduces the average control torque by 1.0 Nm, and highlights its better ability to produce a minimum torque variability of 0.18 Nm, reduce energy use and produce smooth control tasks.

Discussion

The offered ESN-RSA-HHO adaptive control indicated its high ability to track trajectories, energy consumption, and stability of the closed loop in the comparative study of the 2-DOF robotic arm experiments with the standard PID controllers, LSTMs, and unoptimised ESN models. The system was capable of managing nonlinear dynamics and external disturbances, as the RMSE error during its simulation was dramatically reduced to 0.012 rad, and overshoot was minimised to 2.8% with the help of reservoir computing and hybrid metaheuristic optimisation. Diffuse parameters of the ESN converged quite fast, according to the fitness curve and parameter time trajectories in the RSA-HHO hybridisation, achieving an optimal value of the spectral radius, leakage rate, and input scaling

to be dynamic (adaptive). The resulting energy saving was 28% when compared with PID controllers as a result of an increase in the smoothness of the torque signals, and the plots showed the presence of robust stability during disturbed implementation, and the tracking errors also remained bounded around equilibrium. The proposed controller was a lightweight but highly adaptive controller, in contrast to PID, which had to handle nonlinearities, and LSTM, which needed vast amounts of training; thus, the proposed controller worked well in real-time. The results indicate that the ESN-RSA-HHO method is effective in implementing complex mechatronic systems and opens the future direction of future work to include actual hardware implementation, higher-DOF arms and fusing with sensor fusion methods to make an entire system highly robust in real-life applications.

V. CONCLUSION

The proposed study suggests an adaptive control technique of a 2-DOF robotic arm with an Echo State Network (ESN) and optimised with a hybrid algorithm, Harris Hawks Optimisation and Reptile Search Algorithm (HHO-RSA). The framework proposed was capable of handling the problems of nonlinearity and external forces and performed considerably better than the conventional PID controllers and LSTM in terms of trajectory tracking accuracy and energy efficiency, as well as stability. Enhanced ESN parameters produced smooth torque curves and faster convergence that were confirmed by thorough simulations. The obtained results point to the promising nature of the ESN-RSA-HHO method as an effective and energy-saving method of control of complex mechatronic systems. Subsequent work will be aimed at generalisation of the proposed framework to higher-DOF robotic manipulators and physical hardware prototypes in order to prove its efficiency in practical settings, such as non-modelled dynamics, time lag, and inaccurate sensing. Also, sensor fusion methods and adaptive learning abilities may need to be embraced to generate further toughness towards uncertainties and dynamic settings. In order to enhance computational efficiency, parallelisation schemes and surrogate model techniques can be investigated to speed up the RSA-HHO optimisation process. Lastly, the proposed control architecture would be analysed in terms of its possible use in collaborative robotics, exoskeletons, and autonomous systems so that its scalability and applicability could be demonstrated.

References

[1] A. Rodríguez-Molina, M. G. Villarreal-Cervantes, E. Mezura-Montes, and M. Aldape-Pérez, "Adaptive controller tuning method based on online multi-objective optimization: A case study of the four-bar mechanism," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1272–1285, 2019.

[2] A. Mompó Alepuz, D. Papageorgiou, and S. Tolu, "Brain-inspired biomimetic robot control: a review," *Front. Neurorobotics*, vol. 18, p. 1395617, 2024.

[3] L. Chen *et al.*, "An Optimization Method for Multi-Robot Automatic Welding Control Based on Particle Swarm Genetic Algorithm," *Machines*, vol. 12, no. 11, p. 763, 2024.

[4] T. H. Nehal, W. M. Hamanah, and M. A. Abido, "Advancements in Optimization Techniques for Active Magnetic Bearing Systems: Current Trends and Future Directions," *IEEE Access*, 2025.

[5] A. Moloody, A. A. Barry, S. S. Ang, R. Kamil, and A. Zolfagharian, "PID Controller Parameter Tuning Based on a Modified Differential Evolutionary Optimization Algorithm for the Intelligent Active Vibration Control of a Combined Single Link Robotics Flexible Manipulator," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 52, pp. 234–258, 2025.

[6] A. Nuthan *et al.*, "Reinforcement Learning in Mechatronic Systems: A Case Study on DC Motor Control," *Circuits Syst.*, vol. 16, no. 01, pp. 1–24, 2025.

[7] A. Nazari, A. Aghajani, P. Buhr, B. Park, Y. Wang, and C. Shafai, "Using Adaptive Surrogate Models to Accelerate Multi-Objective Design Optimization of MEMS," *Micromachines*, vol. 16, no. 7, p. 753, 2025.

[8] J. F. Rosero, N. Cardozo, and I. Dusparic, "Multi-Objective Deep Reinforcement Learning Optimisation in Autonomous Systems," in *2024 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, IEEE, 2024, pp. 97–102.

[9] J. Mary, U. S. Banu, D. Dinakaran, and R. Nakandhrakumar, "Adaptive control by multi-objective optimisation for drilling process with fuzzy inference system and neural predictive controller," *Insight-Non-Destr. Test. Cond. Monit.*, vol. 59, no. 1, pp. 38–44, 2017.

[10] Z. Fan, Z. Zheng, B. Xu, W. Li, Y. Zhang, and Z. Hao, "Performance optimization of hard rock tunnel boring machine using multi-objective evolutionary algorithm," *Comput. Ind. Eng.*, vol. 169, p. 108251, 2022.

[11] W. Li *et al.*, "Modular design automation of the morphologies, controllers, and vision systems for intelligent robots: a survey," *Vis. Intell.*, vol. 1, no. 1, p. 2, 2023.

[12] Y. Pan, K. Guo, T. Sun, and M. Darouach, “Bioinspired composite learning control under discontinuous friction for industrial robots,” *ArXiv Prepr. ArXiv220612195*, 2022, [Online]. Available: https://arxiv.org/abs/2206.12195
[13] J. Hu and others, “Deep reinforcement learning with LSTM and GAIL for trajectory tracking,” *Biomimetics*, vol. 9, no. 12, pp. x–x, 2024, doi: 10.3390/biomimetics9120XXX.
[14] Y. Zhu and others, “Coordinated locomotion via multi-objective whale optimization,” *Biomimetics*, vol. 10, no. 5, 2025, doi: 10.3390/biomimetics1005XXXX.
[15] S. K. Boddhu and J. C. Gallagher, “Evolving neuromorphic flight control for a flapping-wing mechanical insect,” *Int. J. Intell. Comput. Cybern.*, vol. 3, no. 1, pp. 94–116, 2010.
[16] M. Hiraga and others, “Echo state networks in robotic swarms,” *Unpubl. Ser.*, 2023.
[17] Y. Li, H. Liu, and H. Gao, “Online learning fuzzy echo state network with applications on redundant manipulators,” *Front. Neurorobotics*, vol. 18, p. 1431034, 2024, doi: 10.3389/fnbot.2024.1431034.
[18] Y. F. Tham and D. V. Vargas, “Generating oscillation activity with ESN to mimic CPG behavi[illegible], *ArXiv Prepr. ArXiv230610927*, 2023, [Online]. Available: https://arxiv.org/abs/2306.10927
[19] A. Banderchuk and others, “Combining robust control and machine learning for uncertain nonlinear systems subject to persistent disturbances,” *ArXiv Prepr. ArXiv230311890*, 2023, [Online]. Available: https://arxiv.org/abs/2303.11890
[20] E. Galván and F. Stapleton, “Evolutionary multi-objective optimisation in neurotrajectory prediction,” *ArXiv Prepr. ArXiv230802710*, 2023, [Online]. Available: https://arxiv.org/abs/2308.02710
[21] W. K. Li and others, “Multi-objective evolutionary design of central pattern generator network for biomimetic robotic fish,” *Complex Intell. Syst.*, 2023, doi: 10.1007/s40747-022-00883-7.
[22] S. Basterrech and G. Rubino, “Evolutionary echo state network: A neuroevolutionary framework for time series prediction,” in *Conference on Artificial Evolution*, 2022. [Online]. Available: https://www.researchgate.net/publication/371385607
[23] P. Ngamkajornwiwat, J. Homchanthanakul, P. Teerakittikul, and P. Wanitchanpong, “Bio-inspired adaptive locomotion control system for online adaptation of a walking robot on complex terrains,” *Ieee Access*, vol. 8, pp. 91587–91602, 2020.
[24] J. L. Templos-Santos, O. Aguilar-Mejia, E. Portilla-Flores, and R. Sosa-Cortez, “Parameter tuning of PI control for speed regulation of a PMSM using bio-inspired algorithms,” *Algorithms*, vol. 12, no. 3, p. 54, 2019.
[25] Z. Elgamal, A. Q. M. Sabri, M. Tubishat, D. Tbaishat, S. N. Makhadmeh, and O. A. Alomari, “Improved reptile search optimization algorithm using chaotic map and simulated annealing for feature selection in medical field,” *IEEE Access*, vol. 10, pp. 51428–51446, 2022.
[26] S. S. Elashry, A. Abohamama, H. M. Abdul-Kader, M. Rashad, and A. F. Ali, “A chaotic reptile search algorithm for energy consumption optimization in wireless sensor networks,” *IEEE Access*, vol. 12, pp. 38999–39015, 2024.