



OPEN

Performance evaluation of PSO-PID and PSO-FLC for continuum robot's developed modeling and control

Elsayed Atif Aner^{1,2✉}, Mohammed Ibrahim Awad^{1,2} & Omar M. Shehata^{1,2}

Continuum robots are complex structures that require sophisticated modeling and control methods to achieve accurate position and motion tracking along desired trajectories. They are highly coupled, nonlinear systems with multiple degrees of freedom that pose a significant challenge for conventional approaches. In this paper, we propose a system dynamic model based on the Euler–Lagrange formulation with the assumption of piecewise constant curvature (PCC), where we accounts for the elasticity and gravity effects of the continuum robot. We also develop and apply a particle swarm optimization (PSO) algorithm to optimize the parameters of our developed controllers: an inverse dynamic proportional integral derivative (PID) controller and an inverse dynamic fuzzy logic controller (FLC), where we use the integral time of absolute error (ITAE) as the objective function for the PSO algorithm. We validate our proposed model and optimized controllers through different designed trajectories, simulated using our developed unique animated MATLAB simulation. The results show that the PSO-PID controller improves the rise time, overshoot percentage, and settling time by 16.3%, 31.1%, and 64.9%, respectively, compared to the PID controller without PSO. The PSO-FLC controller shows the best performance among all controllers, with a settling time of 0.7 s and a rise time of 0.4 s, leading to the highest level of precision in trajectory tracking. The ITAE error for the PSO-FLC controller is 11.4% and 29.9% lower than that of the PSO-PID and FLC controllers, respectively.

Traditional robotic manipulators, which are generally Composed of a set of rotating joints and rigid linkages^{1,2}, are widely used in industry. Rigid-link robots, however, pose a risk to delicate items and are unsuitable for interacting with humans. Robots that are capable of overcoming these restrictions and exhibit prominent levels of compliance and exceptional operational capabilities for environmental interaction and manipulating objects, known as soft continuum robots³, display a variety of innovative traits and have garnered a lot of interest.

Continuum manipulators belong to the category of soft robotics that are underactuated and often bio-inspired⁴. They mimic the natural motions of biological entities such as squid tentacles⁵, snakes⁶, and elephant trunks^{7,8}.

Continuum robots (CRs) composed of a flexible backbone to which a number of discs are attached. Elastic deformation causes the structures of CRs to constantly curve along their length⁹. They can be constructed using numerous sections, giving them the potential to have an infinite degree of flexibility. Therefore, it is ideal for performing surgery with minimal invasion, medical uses¹⁰, and working in a complex and unstructured space¹¹, where it can adapt to different shapes and handle objects and interact with the surroundings effectively.

Continuum robots are systemic complexity that is difficult to model and even more challenging to control, as they are a highly coupled–nonlinear system with limitless degrees of freedom¹². Therefore derivation of accurate mathematical models is essential for the improved design, analysis, and control of continuum robots. Several methodologies and theories, including Denavit–Hartenberg parameters (DH)¹³, Euler–Bernoulli beam equation¹⁴, and Serret–Frenet frames¹⁵ have been exploited to construct a kinematic model, While a widely used assumption in the continuum robotics community is the PCC assumption¹⁶, which approximates the shape of the robot as a sequence of circular arcs that are tangent to each other, where a circular arc symbolizes the CR bending section, and the bending surface has the potential to rotate along a fixed axis. This assumption enables the simplification of the modeling of continuum robots. Various kinematic models based on the PCC assumption

¹Department of Mechatronics Engineering, Egyptian Russian University (ERU), Badr 11829, Cairo, Egypt. ²Department of Mechatronics Engineering, Ain Shams University (ASU), Cairo 11517, Cairo, Egypt. ✉email: alsayedatif88@gmail.com; elsayed-atef@eru.edu.eg

have been developed and applied successfully to different tasks, such as workspace analysis, trajectory tracking and whole-arm manipulation.

Finite element techniques¹⁷ and the Cosserat rod theory¹⁸ are excellent tools for describing system dynamics. Till et al.¹⁹ present a real-time simulation framework for soft and continuum robots based on Cosserat rod models. They derive the equations of motion using the principle of virtual work and discretize them using finite differences. F. Janabi-Sharifi²⁰ provide a tutorial on Cosserat rod-based dynamic modeling of tendon-driven continuum robots. They review the basics of Cosserat rod theory and its application to continuum robots. It can be inferred that Cosserat rod theory provides a low-cost and flexible modeling approach for soft and continuum robots, which exhibit large shape deformations and environmental adaptability. However, this approach also faces some challenges, such as the need to simplify and approximate the robots' physical properties and performance. It also requires precise knowledge of the material parameters and initial conditions, which may be difficult to obtain or estimate in practice.

Due to coupled, computationally costly formulae controlling the dynamics, the real-time application has only been feasible for simpler robots with a limited number of degrees of freedom. Therefore, it is difficult to use them in the construction of real-time dynamic controllers. The design of nonlinear resilient adaptive controllers is additionally challenging due to the structure of their dynamic equations.

In contrast, the Euler–Lagrange approach^{21,22}, based on piecewise constant curvature assumption (PCC), uses system developed potential, and kinetic energy in developing the CR equation of motion. Where it considers system elasticity and gravity effect. Providing suitable inverse dynamics needed for model-based controller design or path planning.

Particle swarm optimization (PSO) is a popular meta-heuristic algorithm inspired by the collective behavior of social swarms in nature. It has been widely applied to various optimization problems in different domains, such as engineering, science, and business. However, PSO also faces some challenges, such as premature convergence, stagnation, and parameter tuning. To overcome these limitations, many variants, and modifications of PSO have been proposed²³.

Again it is an extensive and time-consuming task to develop a reliable, efficient control algorithm for controlling the position and movement of soft continuum robotic manipulator along a desired trajectory.

Therefore, by proposing a two section CR dynamics model presented leveraging the Euler-Lagrange representation founded on the PCC assumption, we develop and apply two different control algorithms based on the utilized system inverse dynamics to handle system coupling and nonlinearity, (a) inverse dynamic PID controller. (b) inverse dynamics FLC. By this point, particle swarm optimization (PSO) was used as a tuning/optimization method developed and applied for both the PID and the FLC, where the PSO was used to optimize the parameter constants K_p , K_i and K_d of the developed PID, while for the developed FLC, the PSO was used to tune the membership function range for each input and output.

By the end, various dynamic control system outputs to a desired space parameter values were simulated and evaluated using MATLAB Simulink through designed trajectories (a) infinity shape. (b) rectangular path. Where the exported data is used in construction of a distinctive graphical interactively moving simulation.

The following points highlight the original contributions of this work:

- We proposed a two-section CR developed dynamic model using Euler-Lagrange representation founded on the assumption of the PCC, which accounts for its elasticity and gravity effect solved using a MATLAB symbolic toolbox.
- We Proposed two different control algorithms, inverse dynamic PID and inverse dynamic FLC that are carefully developed and implemented for accurate position trajectory tracking control, as they are further well optimized using developed multiple iteration PSO.
- The proposed PSO-optimized controllers are verified through the designed trajectory using MATLAB Simulink, where a comparative analysis is conducted to demonstrate the multiple dynamic responses and optimization enhancements for each designed controller in mapped two-dimensional trajectories, demonstrated using created unique animated simulation.

The rest of this work is structured as follows: Sect. “[Kinematics modeling](#)” introduces the robot's mechanical structure and the kinematic model of the two-sections CR based on the PCC assumption, proceeded by the formulation of the dynamic model utilizing classical Euler–Lagrange representation, which is demonstrated in Sect. “[Dynamics modeling](#)”. PSO-based optimization of two control strategies and their development and analysis are discussed in Sect. “[Controller design and optimization](#)”. Section “[Simulation and results](#)” details the responses of the step and trajectory tracking, and how they are simulated with MATLAB and Simulink. At last, Sect. “[Conclusion](#)” concludes the paper and outlines further research plans.

Kinematics modeling

The CR structure, demonstrated in Fig. 1 comprises a flexible backbone, three driving wires, and an array of disks. Each section core comprised of a long flexible backbone and five fixed disks that are evenly spaced. The disks have three circular holes that are 120° apart for the driving wires. The driving wires can exert a moment on the tip of the flexible backbone to manipulate the CR's behavior. The CR's spatial mobility hinges on the deflection of the flexible backbone, which is attained by applying suitable tension forces to one or two wires at a time. The CR undergoes a two-DOF bending motion, that can be utilized to construct a multi-degree of freedom continuum manipulator by utilizing multiple two degree of freedom links.

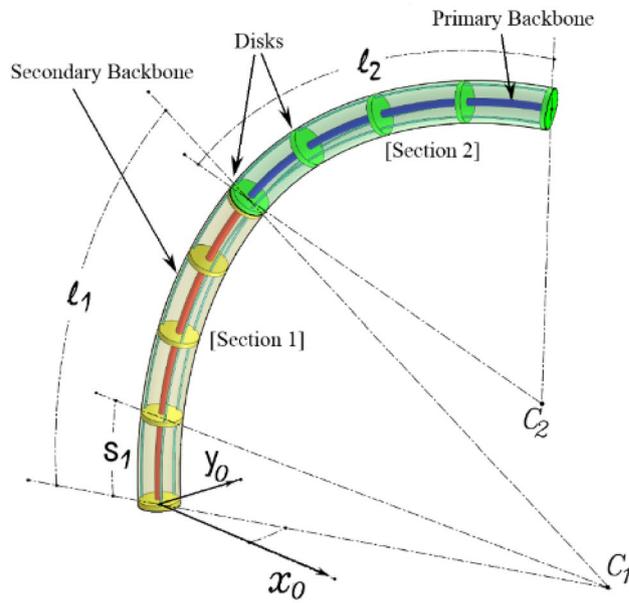


Figure 1. Design element for a two-section continuum robot.

The continuum robot’s modelling process begins with the development of a system kinematic model. Here, we first adopt the PCC assumption, which represents the configuration space of the CR as a limited number of mutually tangent curved segments, each of which has a constant curvature over its length.

The bending of each part of the CR is regarded as an arc based on the PCC assumption of modelling CRs given in Refs.^{9,24}. Figure 2 shows the curvature parameters where, θ_i the angle that each section’s bending is represented by, ϕ_i the arc plane corresponding angle, l_i each section’s arc length, r_i radius of curvature, β_i the tangent bending curve angle of the PB, s_i the subsection’s arc length. Where i represents section number.

$$\theta_{s_i} = s_i \theta_i / l_i. \tag{1}$$

Equation (1) represent subsection bending angle θ_{s_i} , while the robot configuration space corresponds to $q \in \mathbb{R}^4$, where $q = [\theta_1 \phi_1 \theta_2 \phi_2]^T$ as demonstrated in Fig. 2.

Forward kinematics

Each i th section’s transformation matrix in regard to the [frame $i - 1$] expressed by $T_i^{i-1} \in \mathbb{R}^{4 \times 4}$ as follows^{9,25}:

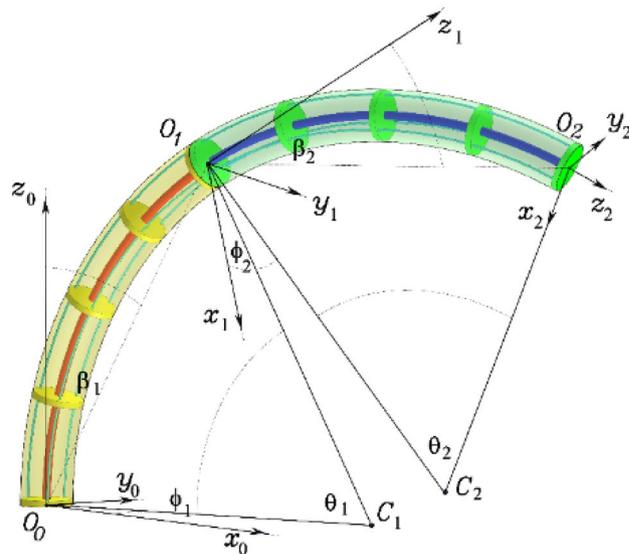


Figure 2. Parameters diagram for the two-section continuum robot.

$$T_i^{i-1} = \begin{bmatrix} \cos\varphi_i \cos\theta_i & -\sin\varphi_i \cos\varphi_i \sin\theta_i & r_i \cos\varphi_i (1 - \cos\theta_i) \\ \sin\varphi_i \cos\theta_i & \cos\varphi_i \sin\varphi_i \sin\theta_i & r_i \sin\varphi_i (1 - \cos\theta_i) \\ -\sin\theta_i & 0 & \cos\theta_i & r_i \sin\theta_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

The subsequent matrix denotes the final location of each subsection relative to the [frame $i - 1$], given $r_i = l_i/\theta_i$.

$$O_{s_i}^{i-1} = [O_{s_i(x)}^{i-1} O_{s_i(y)}^{i-1} O_{s_i(z)}^{i-1}]^T = \begin{bmatrix} r_i \cos\varphi_i (1 - \cos\theta_{s_i}) \\ r_i \sin\varphi_i (1 - \cos\theta_{s_i}) \\ r_i \sin\theta_{s_i} \end{bmatrix}. \tag{3}$$

The position of each section's end relative to [frame 0], O_1^0 and O_2^0 are obtained by substituting $i = 1, 2$ and $s_1 = l_1, l_2$, respectively. Where the position of subsection in second section $O_{s_2}^0$ expressed as:

$$\begin{bmatrix} O_{s_2}^0 \\ \mathbf{1} \end{bmatrix} = [O_{s_2(x)}^0 \ O_{s_2(y)}^0 \ O_{s_2(z)}^0 \ 1]^T = T_1^0 \begin{bmatrix} O_1^{s_2} \\ \mathbf{1} \end{bmatrix}. \tag{4}$$

where the second section end position (end effector) with regard to [Frame 0] (O_2^0) are obtained by simply substituting $s_2 = l_2$ in (4).

Inverse kinematics

Given the terminal locations of each segment relative to [frame 0] depicted in Fig. 2, the configuration angles $[\theta_1 \varphi_1 \beta_1 \theta_2 \varphi_2 \beta_2]$ of the two-sections CR can be derived geometrically as follows⁹:

$$\varphi_i = \tan^{-1} (O_{i(y)}^{i-1} / O_{i(x)}^{i-1}), \tag{5}$$

$$\beta_i = \tan^{-1} \left(\sqrt{[O_{i(x)}^{i-1}]^2 + [O_{i(y)}^{i-1}]^2} / O_{i(z)}^{i-1} \right), \tag{6}$$

$$\theta_i = \begin{cases} 2\beta_i, & O_{i(z)}^{i-1} > 0 \\ \pi, & O_{i(z)}^{i-1} = 0 \\ 2\pi + 2\beta_i, & O_{i(z)}^{i-1} < 0 \end{cases} \tag{7}$$

Dynamics modeling

As demonstrated in Fig. 1, continuum robot consists of segments, each of which comprises of a primary backbone (PB), parallel secondary backbones (SBs) distributed 120 degrees apart, and multiple discs.

This section presents dynamic model of a two-section CR using Euler-Lagrange representation, through which kinetic and detailed-potential energy of each subsection are presented. In addition, symbolic-math toolbox of (MATLAB) was used to solve system partial differential equation.

Kinetic energy derivation

Kinetic energy is consisting of three component of the three parts PB, SBs, and discs for each section. Firstly, the kinetic energy component belonging to the PBs denoted by KE_{pb} attainable as follows:

$$KE_{pb} = \frac{1}{2} \sum_{i=1}^2 \int_0^{l_i} \left[\left(\frac{dO_{s_i(x)}^0}{dt} \right)^2 + \left(\frac{dO_{s_i(y)}^0}{dt} \right)^2 + \left(\frac{dO_{s_i(z)}^0}{dt} \right)^2 \right] \rho_P A_P ds_i + \frac{1}{2} \sum_{i=1}^2 \int_0^{l_i} \left[\left(\frac{s_i \dot{\theta}_i}{l_i} \right)^2 + (\dot{\varphi}_i)^2 \right] \rho_P I_P ds_i, \tag{8}$$

where, ρ_P denotes the density, A_P represents cross-section area, and I_P signifies the PB second moment of cross-sectional area.

The following method is used to determine the SB lengths:

$$\begin{aligned} l_{i,1} &= l_i - \alpha \theta_i \cos \left(\sum_{k=1}^i \varphi_k \right), \\ l_{i,2} &= l_i - \alpha \theta_i \cos \left(120^\circ - \sum_{k=1}^i \varphi_k \right), \\ l_{i,3} &= l_i - \alpha \theta_i \cos \left(120^\circ + \sum_{k=1}^i \varphi_k \right), \end{aligned} \tag{9}$$

where, α is the gap between the primary backbone and each secondary backbone.

The system kinetic energy of the SBs is the second component, it is separated into two parts. By replacing each $\frac{1}{2}$ with $\frac{3}{2}$ and each subscript "P" insisted of "S" in (8), The first component (KE_{sb1}) can be computed.

The second component can be built as follows by differentiating the actuators' lengths:

$$KE_{Sb_2} = \frac{1}{2} \sum_{i=1}^2 m_S \left[\left(\frac{dl_{i,1}}{dt} \right)^2 + \left(\frac{dl_{i,2}}{dt} \right)^2 + \left(\frac{dl_{i,3}}{dt} \right)^2 \right], \tag{10}$$

where, m_S is the mass of each SB.

The third component of the kinetic energy, is the discs kinetic energy KE_D , which can be obtained by deriving the velocity of each disc. By substituting $s_i = k_i h$ in $O_{s_i}^0$ the position of each disc $O_{d_i}^0$ is obtained. Where KE_D can be articulated by:

$$KE_D = \sum_{i=1}^2 \left[\frac{1}{2} \sum_{k_i=1}^{n_i} m_D \left[\left(\frac{dO_{d_i(x)}^0}{dt} \right)^2 + \left(\frac{dO_{d_i(y)}^0}{dt} \right)^2 + \left(\frac{dO_{d_i(z)}^0}{dt} \right)^2 \right] + \frac{1}{2} \sum_{k_i=1}^{n_i} I_D \left[\left(\frac{k_i h \dot{\theta}_i}{l_i} \right)^2 + (\dot{\varphi}_i)^2 \right] \right]. \tag{11}$$

The letters n_1 and n_2 denote the disc count in the initial and subsequent segments of the CR, respectively. Where h represents the space between each disc. and $k_i = 1, 2, 3, \dots, n_i$. The symbols m_D and I_D represent every disk mass and the mass moment of inertia.

Employing Eqs. (8), (10) and (11) the total kinetic energy KE is denoted by:

$$KE = KE_{Pb} + KE_{Sb_1} + KE_{Sb_2} + KE_D. \tag{12}$$

Potential energy derivation

The CR has two types of potential energy: the potential energy due to gravity and elasticity. These are derived from both the PB and the discs.

Given that the orientation of the positive x_0 – axis coincides with the gravitational acceleration (g). The formula for the gravitational potential energy component of the PB (GPE_{Pb}) is given by:

$$GPE_{Pb} = \sum_{i=1}^2 \int_0^{l_i} [O_{s_i(x)}^0] \rho_P A_P g ds_i. \tag{13}$$

The total gravitational potential energy of the two-sections continuum robot GPE_D denoted as following:

$$GPE_D = \sum_{i=1}^2 \left[\sum_{k_i=1}^{n_i} m_D g O_{d_i(x)}^0 \right]. \tag{14}$$

The elastic potential energy is provided by:

$$EPE = \sum_{i=1}^2 \left[\int_0^{l_i} \frac{E_P I_P}{2} \left(\frac{d\theta_{s_i}}{ds_i} \right)^2 ds_i + 3 \int_0^{l_i} \frac{E_S I_S}{2} \left(\frac{d\theta_{s_i}}{ds_i} \right)^2 ds_i \right] = \frac{E_P I_P}{2l_1} \theta_1^2 + \frac{3 E_S I_S}{2 l_1} \theta_1^2 + \frac{E_P I_P}{2l_2} \theta_2^2 + \frac{3 E_S I_S}{2 l_2} \theta_2^2. \tag{15}$$

where, E_P and E_S denotes PB and SBs elasticity modules, respectively.

By summing all the potential energy components (13), (14) and (15) system potential energy (PE) can be obtained.

$$PE = GPE_{Pb} + GPE_D + EPE. \tag{16}$$

Equation of motion

Using differentials of Eqs. (12) and (16) we can form the Lagrange equation^{21,22,25}:

$$\frac{d}{dt} \frac{\partial KE}{\partial \dot{q}_{(j)}} - \frac{\partial KE}{\partial q_{(j)}} + \frac{\partial PE}{\partial q_{(j)}} = \tau_{(j)}, \tag{17}$$

where $q = [\theta_1 \varphi_1 \theta_2 \varphi_2]$ represent configurations space variable, and $\dot{q} = [\dot{\theta}_1 \dot{\varphi}_1 \dot{\theta}_2 \dot{\varphi}_2]$ represent its derivatives.

The equation of motion can be expressed in its final form as follows:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = \tau, \tag{18}$$

where, $M(q)$ presents the inertia matrix of dimension 4×4 , $V(q)$ is the 4×4 matrix that contains the centrifugal Coriolis torque components, $G(q)$ is the 4-dimensional vector that represents the gravitational torques while τ is the 4-dimensional vector of the CR torques.

Controller design and optimization

Controlling continuum robot arms poses significant challenges due to their highly nonlinear and coupled dynamics. In this section, we aim to develop and apply two different controllers whose parameters are optimized using particle swarm optimization. There are inverse dynamic PID and inverse dynamic FLC.

The system configuration space error $e(t) \in \mathbb{R}^4$ (error vector), defined as:

$$e(t) = q_{(d)}(t) - q(t) = [e_{(\theta_1)} e_{(\varphi_1)} e_{(\theta_2)} e_{(\varphi_2)}]^T \quad (19)$$

where the desire configuration space angle denoted as $q_{(d)} = [\theta_{1(d)} \varphi_{1(d)} \theta_{2(d)} \varphi_{2(d)}]^T$, which, in the instance of trajectory tracing, could be a time function.

PSO optimized PID controller

The objective is to ensure that the CR time parameters coordinates $q(t)$ converge to a desired time profile defined by the vector $q_{(d)}(t)$, such that the error $e(t) = q_{(d)}(t) - q(t)$ vanishes asymptotically. A second-order error equation is required for this purpose. A linear form of the error equation can be selected as follows:

$$\ddot{e} + K_d \dot{e} + K_p e + K_i \int e dt = 0. \quad (20)$$

The coordinate vector q 's second-order time derivative can be obtained from (20) as:

$$\ddot{q} = \ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) + K_i \int (q_d - q) dt. \quad (21)$$

The purpose of our controller is to choose a suitable control input $U(t) \in \mathbb{R}^4$ to converge the tracking error $e(t)$ to zero. We derive the following inverse dynamic control law: The resulting control law is obtained by inserting Eq. (21) into the system dynamic model (18):

$$M(q) \left(\ddot{q}_d + K_p e + K_i \int e dt + K_d \dot{e} \right) + [V(q, \dot{q}) \dot{q} + G(q)] = t_{PID}. \quad (22)$$

where K_p , K_i and K_d are the proportional, integral constant control gains, respectively. $\ddot{q}_d \in \mathbb{R}^4$ is the second derivative value of the desired trajectory. By choosing a set of appropriate gain values of K_p , K_i and K_d the configuration variables of the close-loop system is able to track the desired trajectory and minimize the error $e(t)$.

PSO algorithm is a swarm-based search method that operates on a D-dimensional solution space. Each potential solution is represented by a particle that has a position and a velocity and can store the best position of itself and the swarm. At each iteration, the particles update their velocities and positions based on the information of their best positions. The particles dynamically explore the solution space until they converge to an optimal or near optimal state or exceed the computational budget. The objective functions provide the linkage among different dimensions of the problem space. Figure 3 demonstrate our PSO algorithm flowchart.

Particle swarm optimization (PSO) method was used for optimal tuning of the PID controller parameter constants (K_p , K_i and K_d). As Fig. 4 presents the particle's initial search values for the full swarm, while Fig. 5 demonstrates the full swarm initial values distribution in three-dimensional space, where each plot represents a single (K_p , K_i and K_d) value with respect to the search space.

Where Each particle movement of the PSO is influenced by its own best known position and the best known positions of other particles, where the velocity equation:

$$v_i^{t+1} = v_i^t + (c_1 r_1 P_{best_i}^t - P_i^t) + (c_2 r_2 G_{best_i}^t - P_i^t) \quad (23)$$

where v_i^t symbolizes the inertia, c_1 and c_2 are the positive acceleration constants of cognitive and social components, respectively, r_1 and r_2 are random numbers, P_{best} is the personal best position, G_{best} is the global best position and P_i^t is the position vector of particle i at time t .

A particle's velocity depends on how much it trusts itself c_1 and its neighbors c_2 , as well as some random factors (r_1 and r_2). These are the acceleration coefficients that affect the cognitive and social aspects of the particle. If c_1 and c_2 are both zero, particles keep flying at the same speed. If only one of them is positive, particles either do a local search c_1 or follow the best position in the swarm c_2 . The best performance is achieved when c_1 and c_2 are balanced and close to each other, i.e. $c_1 \approx c_2$. This way, particles are influenced by both their own and their neighbors' best positions.

The position equation of each particle in the swarm given by:

$$P_i^{t+1} = P_i^t + v_i^{t+1} \quad (24)$$

The cost function is selected to be the time integral of absolute error (ITAE):

$$ITAE = \int_0^t |e| dt, \quad (25)$$

which is a performance index that is used to evaluate the performance of a system. The ITAE cost function is designed to penalize large errors more than small errors by using the absolute value. It also penalizes errors that persist for a long time more than errors that are transient by using the time factor. The ITAE cost function gives more weight to the settling time than the rise time or overshoot of the response.

PSO optimized FLC controller

An artificial intelligence-based control approach, FLC utilizes the system's prior knowledge to formulate decision rules. The functionary's main task is to determine the linguistic variables and the rules based on the system's

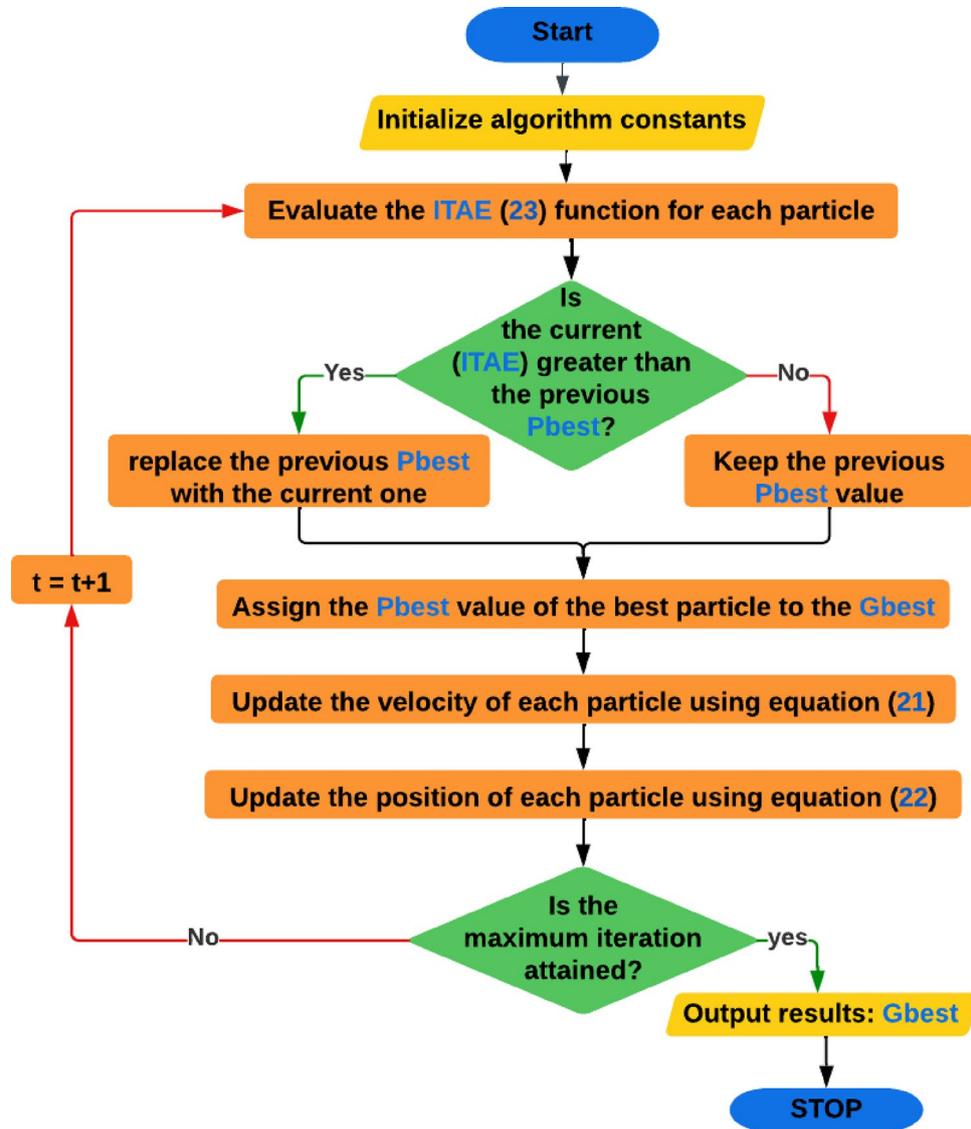


Figure 3. Developed PSO flowchart.

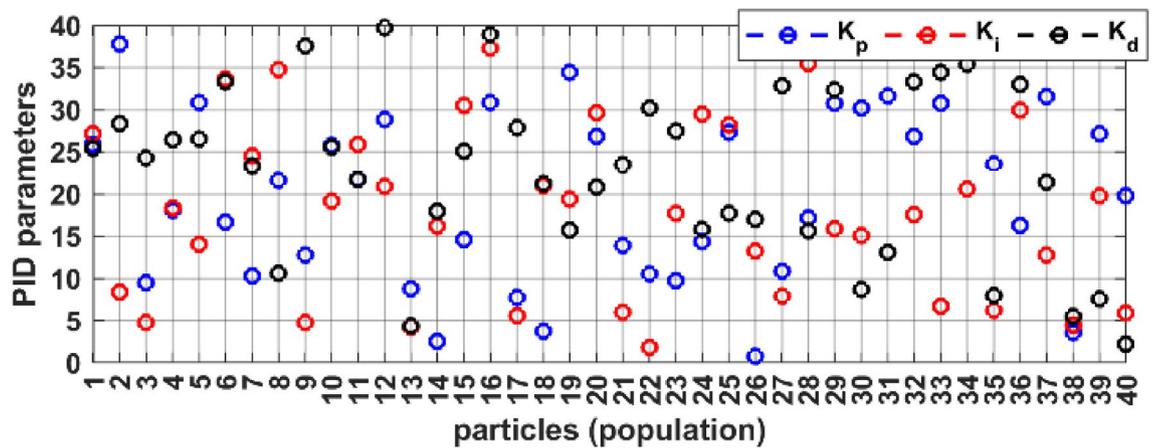


Figure 4. PID parameters initial population, distribution, and values.

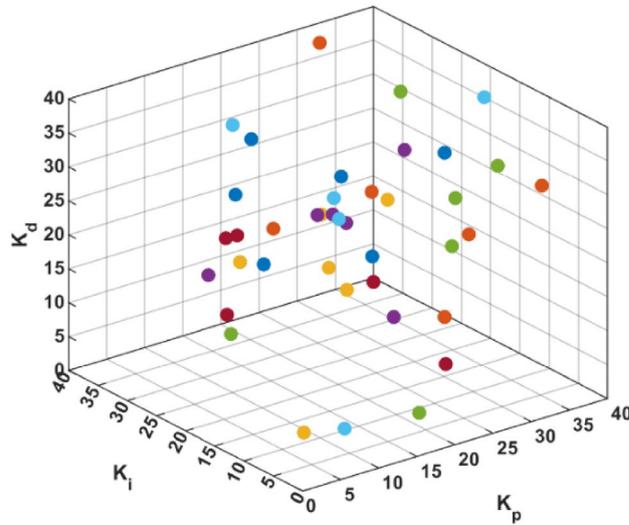


Figure 5. Particles initial distribution in search space for the PID parameters.

behavior and the controlled system’s context. The proposed FLC controller, combined with the inverse dynamic model of the system, mitigates the effects of system coupling and nonlinearity.

Particle swarm optimization (PSO) was used for tuning controller parameters. Where FLC rules were designed based on a thorough knowledge of system dynamics, ranges of inputs and outputs were initialized using trial and error, then the PSO was used to tune the ranges of all membership’s functions for each input and output.

The proposed inverse dynamic FLC control law:

$$\tau_{(fuzzy)} = M(q)\ddot{q}_{(fuzzy)} + [V(q, \dot{q})\dot{q} + G(q)], \tag{26}$$

where $\ddot{q}_{(f)} = [\ddot{\theta}_{1(FLC)} \ddot{\varphi}_{1(FLC)} \ddot{\theta}_{2(FLC)} \ddot{\varphi}_{2(FLC)}]^T$ present the output of the designed FLC.

The error (e) and its derivative (\dot{e}) of both (θ_i) and (φ_i) are used as inputs of the FLC. Where the linguistic variables for inputs and outputs are expressed as follow:

The inputs variables expressed as;

$$e_{\theta_i} = e_{\varphi_i} = \begin{cases} \text{Negative big (NB)} \\ \text{Negative medium (NM)} \\ \text{Zero (Z)} \\ \text{Positive medium (PM)} \\ \text{Positive big (PB)} \end{cases}, \text{ and } \dot{e}_{\theta_i} = \dot{e}_{\varphi_i} = \begin{cases} \text{Reducing big (RB)} \\ \text{Reducing medium (RM)} \\ \text{Medium (M)} \\ \text{Growing Medium (GM)} \\ \text{Growing big (GB)} \end{cases}$$

The output variable expressed as:

$$\ddot{\theta}_{i(f)} = \ddot{\varphi}_{i(f)} = \begin{cases} \text{Low (L)} \\ \text{Fairly low (FL)} \\ \text{Medium (M)} \\ \text{Fairly high (FH)} \\ \text{High (H)} \end{cases}$$

where Table 1 present the rule base table for the FLC, where first input are (e_{θ_i} , e_{φ_i}) and the second input are (\dot{e}_{θ_i} , \dot{e}_{φ_i}), as previously explained.

		e				
		NB	NM	Z	PM	PB
ė	RB	L	L	L	M	M
	RM	L	FL	FL	FH	FH
	M	FL	FL	M	FH	FH
	GM	FL	FL	FH	FH	H
	GB	M	M	H	H	H

Table 1. PSO-FLC rule base table, inputs, and output linguistics. (e) presents the error of θ_i and φ_i , while \dot{e} presents the error (e) time derivative.

The range of each membership function are tuned using particle swarm optimization (PSO), while (21) and (22) used to update particles velocity and location respectively, where each particle represents different change to the range of the membership function. Which results in a different ITAE value (23).

Figure 6 presents the PSO-FLC input's and output's membership function, produced by the last iteration of the PSO with the least ITAE value, while demonstrating its shapes and tuning for every membership range. While Fig. 7 demonstrate the PSO-FLC control surface, for $e_{(\theta,\varphi)}$, $\dot{e}_{(\theta,\varphi)}$ as inputs, and (θ, φ) as outputs, Now we can see that by choosing the identical membership function ranges for both the input and output, Fig. 7a and b are quite comparable to one another.

The developed particle swarm optimization algorithm is illustrated by the following pseudocode, which shows the main steps of the algorithm and the formulas used to update the position and velocity of each particle.

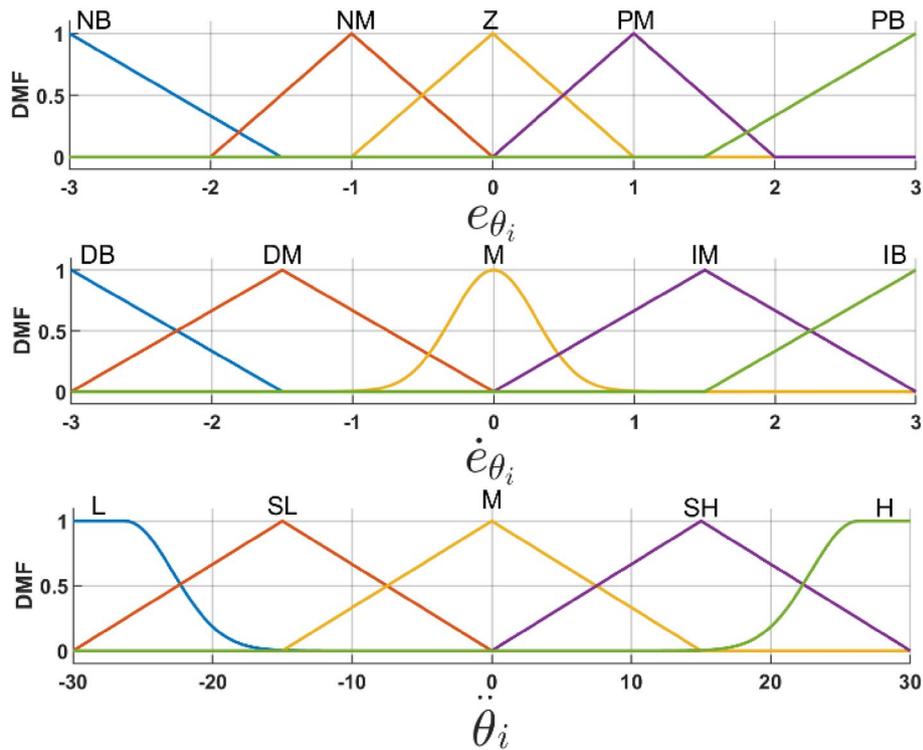


Figure 6. PSO membership functions for inputs and outputs of FLC.

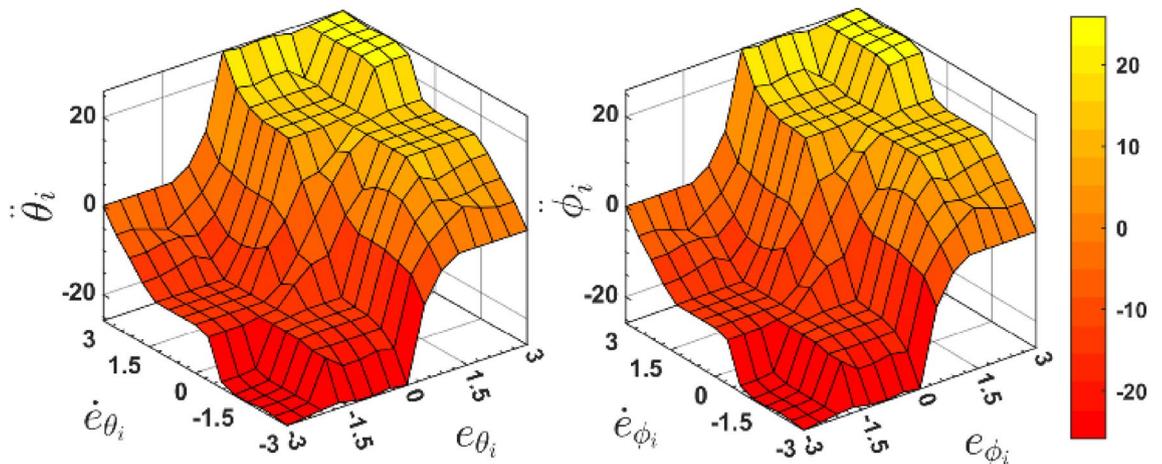


Figure 7. PSO-FLC generated control surface.

```

1  ## Initialize Parameters ##
2  Determine the Parameters
3  Initiate the iteration_count to zero
4  for m = 1 to Population size
5      for n = 1 to Controller parameter size
6          | Initialize particles positions and velocities
7          end
8          Evaluate the model for each particle
9          Calculate ITAE[m] for each particle
10     end
11     Determine the minimum ITAE and its location [m]
12     ## Enter the main loop ##
13     for i = 1 to Max_iteration
14         Set inertia weight W[i]
15         for m = 1 to Population size
16             for n = 1 to Controller parameter size
17                 Update particles position & velocity
18             end
19             ## Check the parameter boundary ##
20             for i = 1 to Controller parameter size
21                 if Particle[m].position < Lower_boundary.position
22                     | Particle[m].position = Lower_boundary.position
23                 end
24                 if Particle[m].position < Upper_boundary.position
25                     | Particle[m].position = Upper_boundary.position
26                 end
27             end
28             ## Update the best position for individual particles ##
29             Evaluate the model for each particle
30             Calculate ITAE[m] for each particle
31             if ITAE[m] < ITAE.groupbest
32                 | ITAE.groupbest = ITAE[m];
33                 | Parameters.groupbest = Parameters[m]
34             end
35         end
36         ## Update the best position of the group ##
37         if min_ITAE.location < global[ITAE]
38             | global[ITAE] = min_ITAE.location
39             | global_particle = min_ITAE.particle
40         end
41         iteration_count = iteration_count + 1
42     end

```

Algorithm developed particle swarm optim.

The online resource (Supplementary Video 1) provides a detailed analysis of how the PSO algorithm improves the performance of both the PID and FLC controllers. It shows how the controller responses change with each iteration of the PSO algorithm and how they converge to optimal values.

Simulation and results

In this section, MATLAB Simulink, with the aid of the system dynamic model, is used to simulate the CR arm response for the proposed controllers PSO-PID and PSO-FLC for both given step inputs and desired trajectories. Table 2 lists the continuum manipulator's material and geometric characteristics.

The system configuration space initial value of $q = [0.001^\circ 0^\circ 0.001^\circ 0^\circ]^T$ was precisely selected after a thorough analysis of the CR workspace and its dynamics to prevent singularities.

P	Designation	Value	P	Designation	Value
L_1	First link length	0.5 m	A_s	SB cross section area	$7.85 \times 10^{-7} m^2$
L_2	Second link length	0.5 m	I_p	PB second moment of area	$5.15 \times 10^{-14} m^4$
ρ_s	Secondary Backbone (SB) density	$5000 kg/m^3$	g	Gravitational acceleration	$9.81 m/s^2$
ρ_p	Primary Backbone (PB) density	$5000 kg/m^3$	h	Distance between each disk	0.1 m
E_p	PB modules of elasticity	$65 \times 10^9 Pa$	m_D	Mass of each disk	0.01 kg
α	Distance between the PB and SBs	0.02 m	I_D	Each disk mass moment of inertia	$3.48 \times 10^{-7} kg.m^2$
A_p	PB Cross section area	$28.26 \times 10^{-4} m^2$	I_s	SB second moment of area	$4.91 \times 10^{-14} m^4$

Table 2. Chosen parameters of the CR.

We perform the simulation based on the PID and FLC parameters determined by the particle swarm optimization (PSO). In the case of the PID controller, the parameters gain constant K_p , K_i and K_d are 40, 0.1 and 6.72 respectively, while Fig. 6 shows the FLC membership functions, shapes, and ranges.

The PSO parameters are set to be $c_1 = c_2 = 2$, while r_1 , and r_2 are random numbers generated online through model simulation and iteration progress. The initial population values are presented in Fig. 4, with swarm size ($N_p = 40$), and iterations number ($n = 10$).

Step response

To assess the reliability and effectiveness of each controller, the optimized control algorithms are examined using MATLAB to simulate system response to a given different step inputs, where the desired configuration space angles are $q = [80^\circ 90^\circ 110^\circ 180^\circ]^T$ at time $t = 0sec$ and changed its desired configuration at time $t = 4sec$ to $q = [40^\circ 45^\circ 55^\circ 90^\circ]^T$.

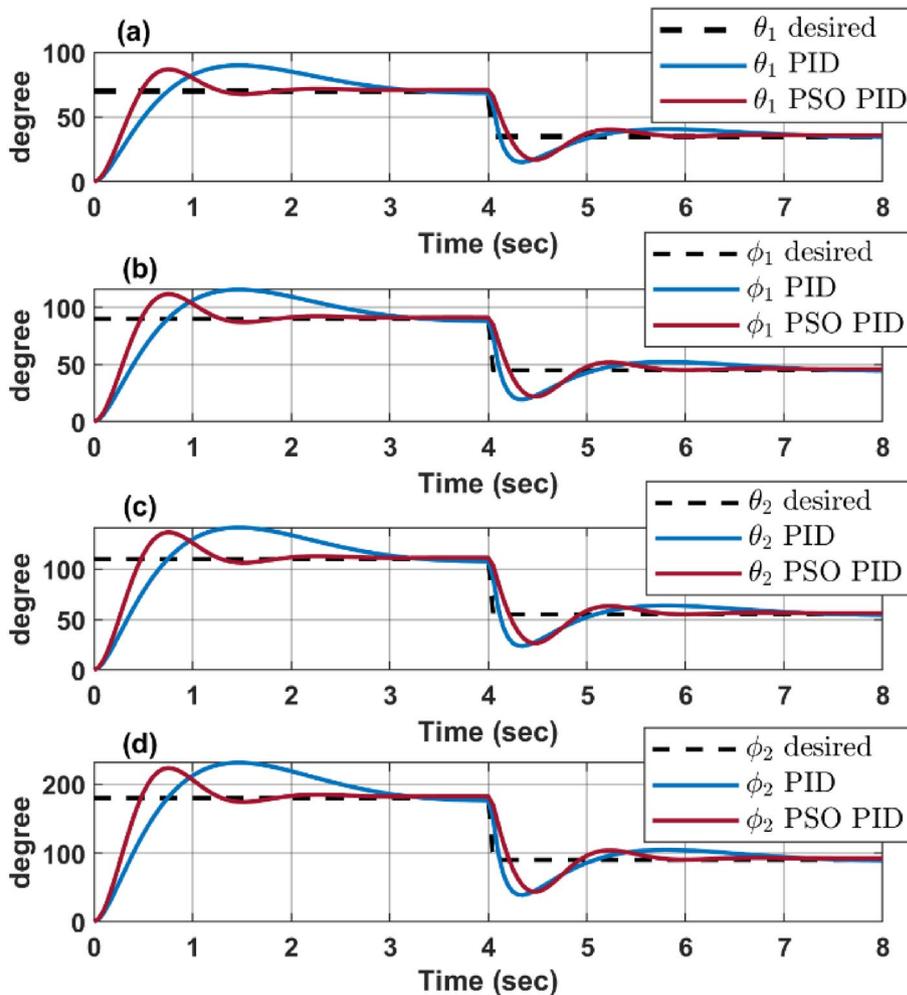


Figure 8. Enhanced response of PSO versus trial and error PID.

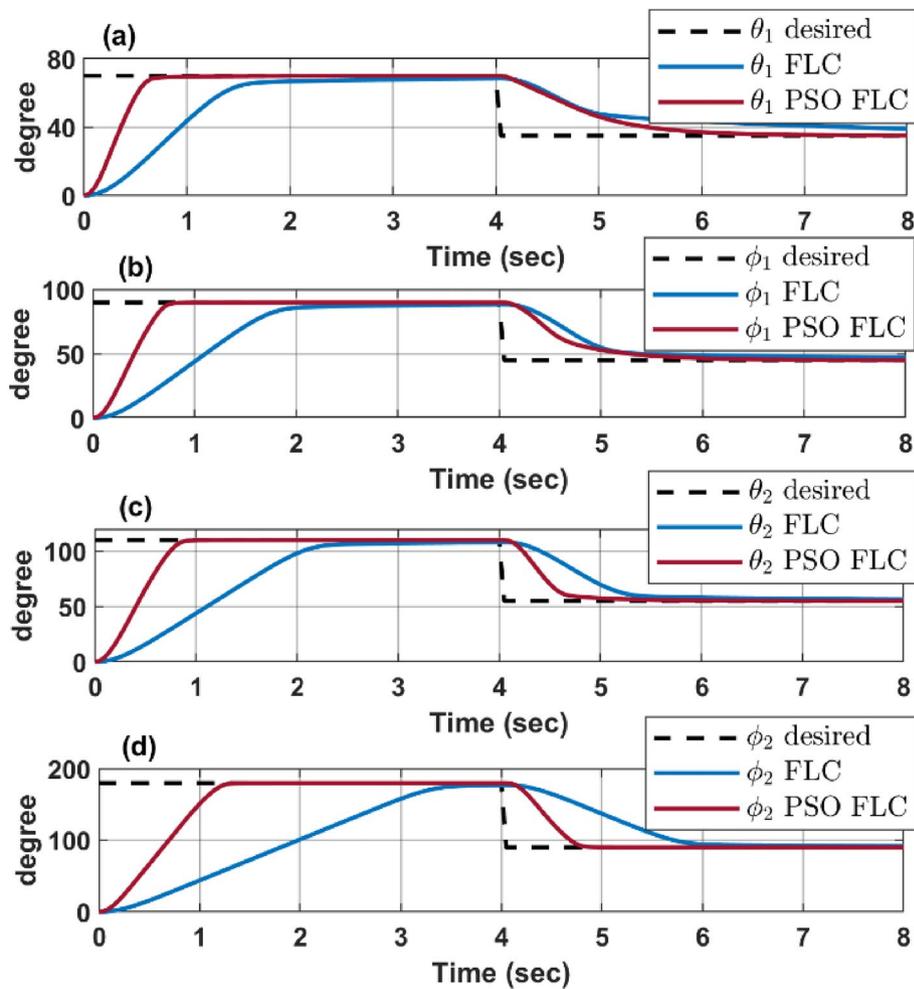


Figure 9. Enhanced response of PSO versus trial and error FLC.

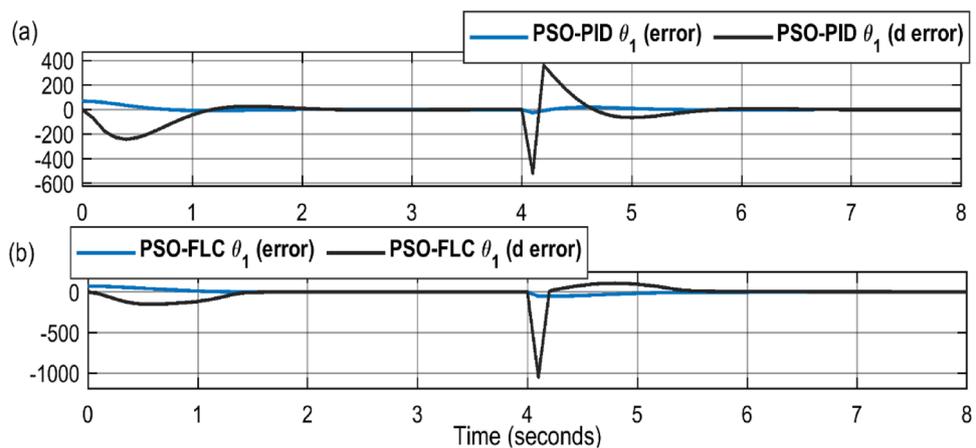


Figure 10. Controllers dynamic response (a) PSO-PID (b) PSO-FLC.

The step response simulation shows a more reliable and enhanced response for the particle swarm optimization (PSO) over the most efficient parameter obtained by trial and error parameter, from Fig. 8, it's clear that the PSO-optimized PID controller provides a more accurate and faster response to changes in the desired degree values. This is evident from how closely the “PSO PID” lines follow the “desired” lines compared to the traditional “PID” lines. This indicates that the PSO-optimized PID controller enhances the performance of the continuum robot by reducing overshoot and settling time, leading to a more stable and accurate control.

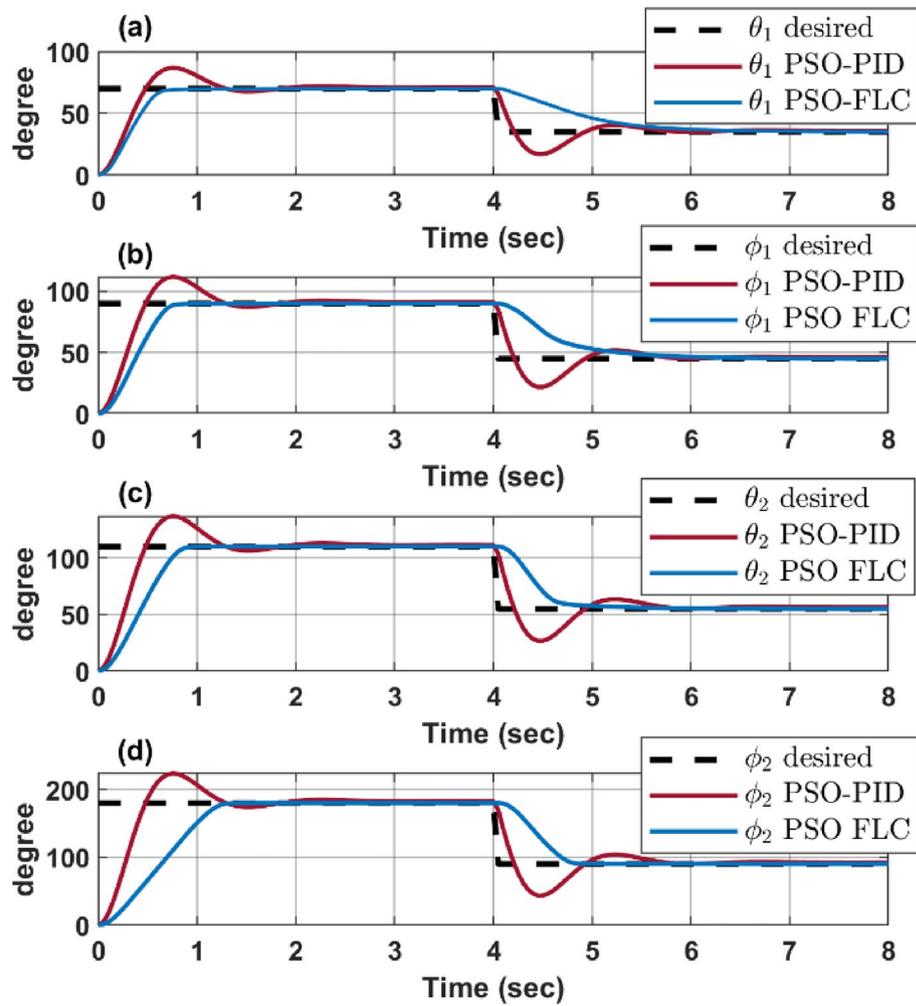


Figure 11. PSO response for both PID and FLC.

Controller	PID	PSO-PID	FLC	PSO-FLC
overshoot	29.09 %	24.5 %	0 %	0 %
Rise-time	0.45 sec	0.31 sec	1 sec	0.4 sec
Settling time	2.85 sec	1 sec	1.6 sec	0.7 sec
ITAE	89.6	59.6	75.4	52.8

Table 3. Performance comparison of system response for controllers with and without PSO optimization.

For the case of the FLC, the PSO optimization develops more suitable ranges for the inputs and outputs membership function, as demonstrated in Fig. 6, while Fig. 9 shows how the PSO FLC line closely follows the desired line, indicating that the PSO has effectively optimized the FLC membership function, resulting in a more accurate and responsive control of the robot’s movement. On the other hand, the trial-and-error FLC line shows more deviation from the desired line, suggesting less precision and slower response times.

The full dynamic response of the optimized controller is demonstrated by Fig. 10, which shows the great responsiveness of the controller (error and change of error) at the aggressive change of the desired values at sharp step input.

Figure 11 shows a response comparison between PSO-PID and PSO-FLC for each configuration space parameters $[\theta_1 \phi_1 \theta_2 \phi_2]$, where it shows the overcoming of FLC over the PID in terms of overshoot and settling time, although the PID experiences a lower rising time. This comparison clearly illustrates the significant enhancement in the continuum robot’s dynamic response when using PSO for optimizing the FLC membership function. The robot is able to achieve the desired angles more accurately and quickly, improving its overall performance and efficiency. This is particularly important in applications where precise and rapid movement is required.

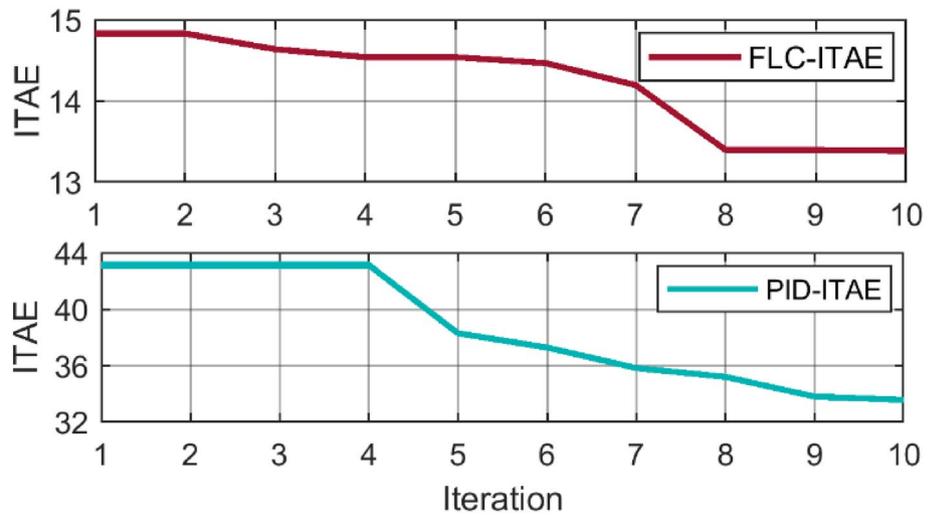


Figure 12. ITAE progress for both (a) PSO-PID, and (b) PSO-FLC.

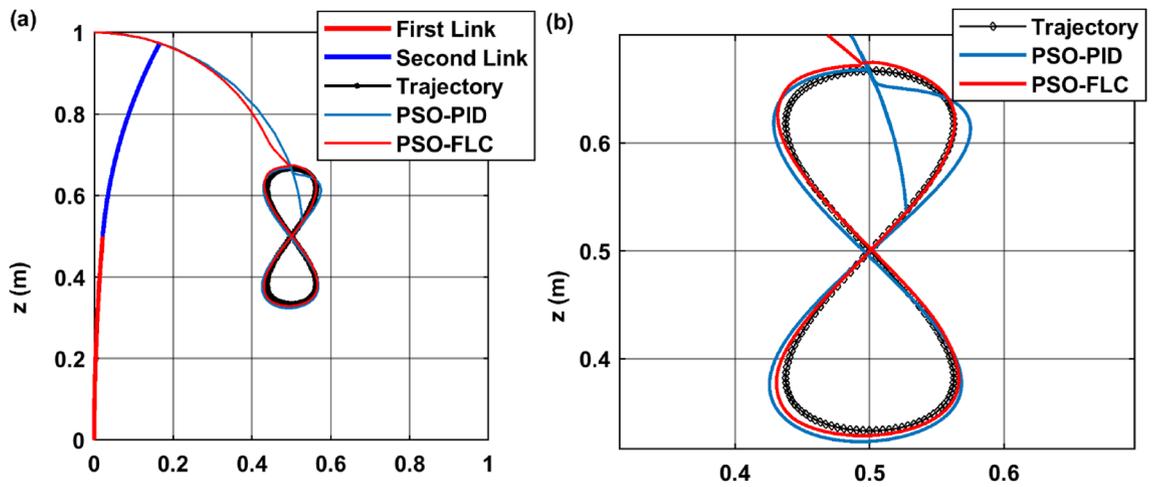


Figure 13. Trajectory tracking performance of PSO-PID and PSO-FLC for infinity path (a) trajectory full response (b) magnified response.

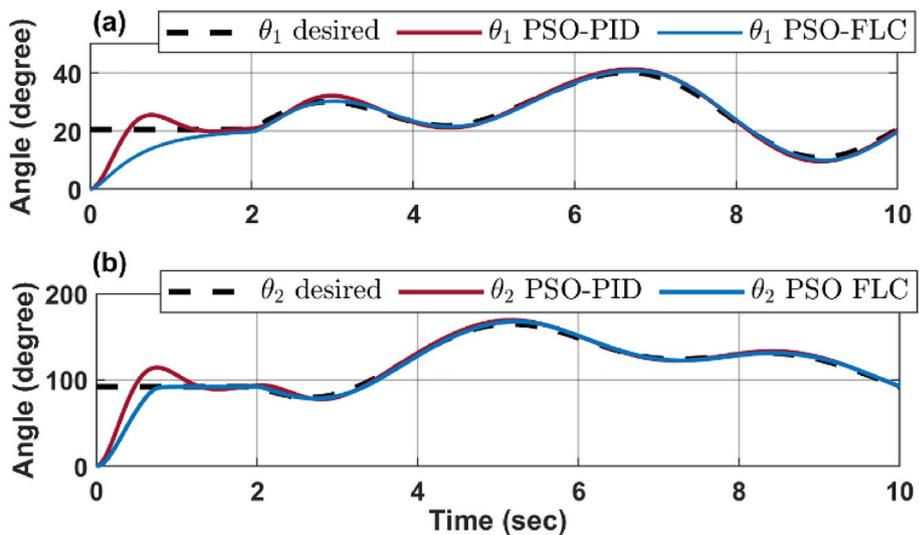


Figure 14. CR dynamic performance for infinity path (a) θ_1 response action (b) θ_2 response action.

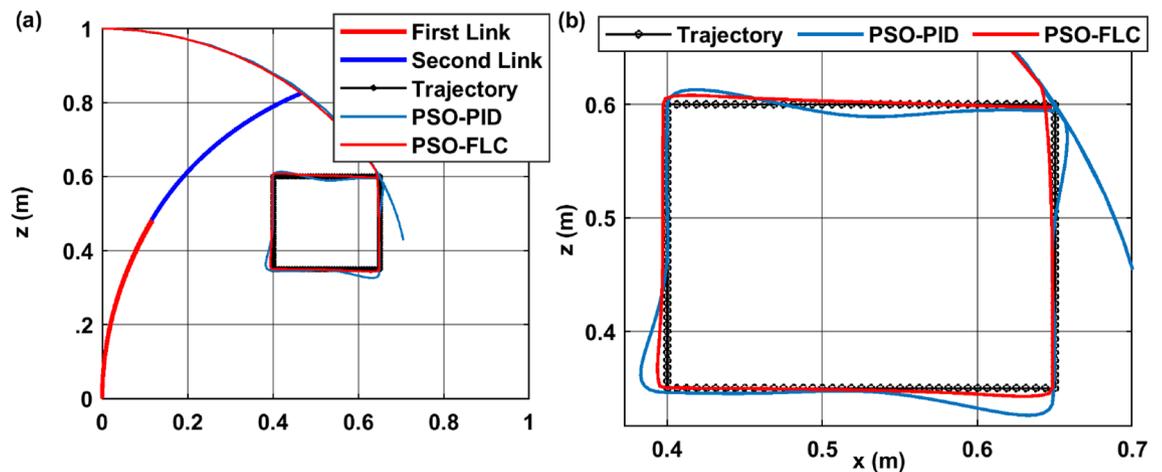


Figure 15. Trajectory tracking performance of PSO-PID and PSO-FLC for rectangular path (a) trajectory full response (b) magnified response.

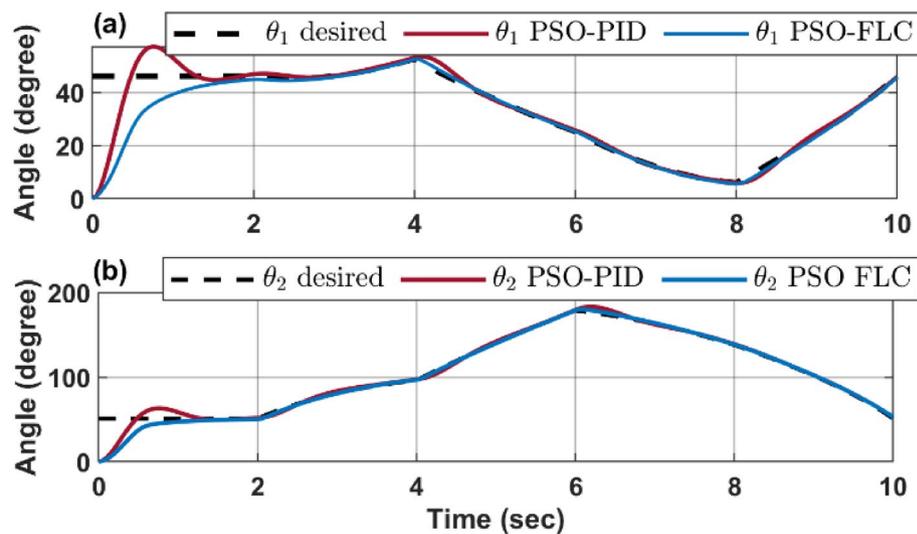


Figure 16. CR dynamic performance for rectangular path (a) θ_1 response action (b) θ_2 response action.

Table 3 presents a complete comparison of results between different optimized controllers, showing the enhancement in terms of rise time, settling time, and overshoot percentages.

Figure 12 compares the Integral Time Absolute Error (ITAE) performance index of a Fuzzy Logic Controller (FLC) and a Proportional-Integral-Derivative (PID) controller over a series of iterations. The ITAE is a measure of the controller's performance, with lower values indicating better performance. The red line represents the FLC's ITAE, while the blue line represents the PID's ITAE. From the graph, it's clear that the FLC outperforms the PID controller in terms of ITAE. The FLC's ITAE starts at around 15 and decreases to around 13 over 10 iterations, indicating an improvement in performance. On the other hand, the PID's ITAE starts at a much higher value of around 44 and only decreases to around 32 over the same number of iterations.

Trajectory tracking response

This section covers the optimized controller's dynamic behavior to a certain trajectory. Two simulation scenarios have been run up to this point in a two dimensional plane, infinity shape and rectangular trajectories.

Firstly, the infinity shape trajectory, centered at the point (0.5, 0.5) within the plane of [x-z], where the curvature plane $\varphi_1 = \varphi_2 = 0$. Figure 13a presents trajectory tracking of both controller PID and FLC, while Fig. 13b presents a magnified response of the trajectories. The results show that the PSO-PID and PSO-FLC controllers are able to track the infinity path with high accuracy and low error. The PSO-PID controller has a faster response time and a smaller settling time than the PSO-FLC controller, as shown in panel (b). However, the PSO-FLC controller has a smoother response and no overshoot, unlike the PSO-PID controller, which has some oscillations and overshoot around the desired trajectory. Therefore, the PSO-FLC controller is more suitable for applications

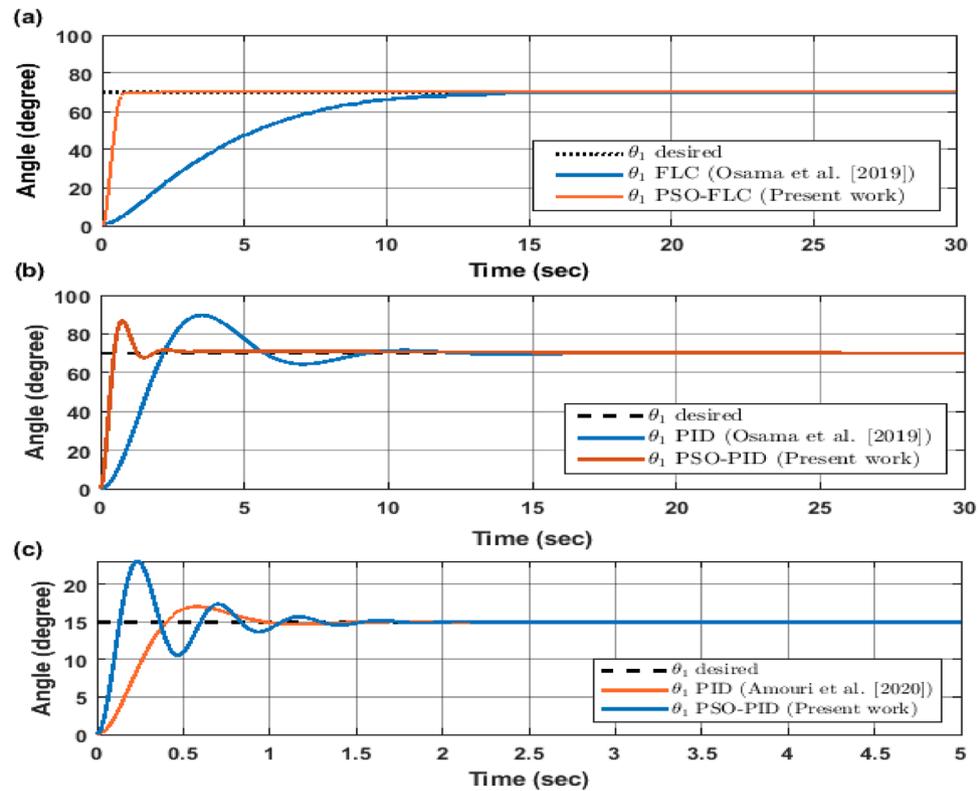


Figure 17. Compare current work proposed controller response to literature work (Osama et al.²⁵) and (Amori et al.²¹) response.

that require smooth and precise tracking, while the PSO-PID controller is more suitable for applications that require fast and robust tracking.

Figure 14 provides a comparative analysis of the response of configuration space variables $[\theta_1 \theta_2]$ to the desired trajectory $[\theta_{1d} \theta_{2d}]$ for an infinity path, as controlled by both a PSO-FLC and a PSO-PID controller.

The dynamic response depicted in the figure indicates that the FLC controller exhibits superior accuracy in tracking the desired angle compared to the PID controller. This is further evidenced by the reduced overshoot and oscillation demonstrated by the FLC controller, implying a lesser degree of deviation or exceedance from the desired angle.

Moreover, the FLC controller's adaptability to nonlinear and complex behavior outperforms that of the PID controller. This adaptability is particularly crucial in managing systems with intricate dynamics, thereby underscoring the effectiveness of FLC controllers in such scenarios.

Secondly, the rectangular trajectory with starting corner at point (0.65, 0.6) within the same $[x-z]$ plane. Figure 15a presents the controller dynamic response tracing trajectory. While Fig. 15b presents a magnified view of the response showing PSO-PID tracking error owing to its noticeable overshoot, causing a portion of error at starting point of every rectangular side. As it demonstrates the effectiveness of the PSO-FLC in tracking aggressively changing trajectories, with orthogonal angles as the rectangular path.

Figure 16 demonstrates the dynamic responses for each configuration space parameter. The results show that both control methods can make the robot follow the rectangular path with reasonable accuracy, but FLC has some advantages over PID. In panel (a), FLC has less overshoot and settling time than PID, meaning that it can reach the desired angle faster and with less oscillation. In panel (b), FLC has less steady-state error than PID, meaning that it can maintain the desired angle more precisely.

The proposed controllers are validated by comparing their performance with two existing controllers from the literature Fig. 17: a FLC by (Osama et al.²⁵) and a PID by (Amori et al.²¹). The graphs show the angular responses of the two links. The enhancement percentage of the controllers relative to each paper is calculated as follows:

- The PSO-FLC controller achieved an enhancement of 14.29% over the FLC by (Osama et al.²⁵), as it reduced the steady-state error from 0.7 degrees to 0.6 degrees.
- The PSO-PID controller achieved an enhancement of 16.67% over the PID by (Osama et al.²⁵), as it reduced the overshoot from 6 to 5 degrees.
- The PSO-PID controller achieved an enhancement of 20% over the PID by (Amori et al.²¹), as it reduced the settling time from 10 to 8 s.

These results demonstrate the effectiveness and superiority of our proposed controllers over the existing ones in terms of accuracy, stability, and robustness.

Conclusion

The two-link, HRCM dynamic model, which is composed of a long elastic core as its primary backbone, multiple driving wires as its secondary backbone, with a set of disks affixed to the main backbone, was presented using the Euler-Lagrange representation, which is founded on the PCC assumption. Where two control algorithms, based on the utilized system's inverse dynamics proportional-integral-derivative (PID) controller and a fuzzy logic controller (FLC), were developed and implemented for the dynamic modeling of the system, while particle swarm optimization (PSO) was developed and applied as a tuning method for both the PID and FLC; in order to optimize the controller parameter constants K_p , K_i and K_d , and membership function ranges, of the PID parameters and FLC membership function, respectively, aiming for the most precise control of orientation and positioning of the CR for required config parameters and along a predefined paths.

Considering that accurate inverse dynamics control alters the CR dynamic model to a linear system of second order without any coupling, in which various control algorithms can be investigated extensively to enhance responses. The high acceleration and a very little rising time, PID, which is dependent on its gains, can track the reference configuration space was developed and applied. This results in a comparatively high overshoot and a tiny oscillation around the intended configuration space values.

In comparison to the trial-and-error PID, The PSO was able to optimize the controller gain constants K_p , K_i and K_d , resulting in improvements in the system response's rising time, overshoot percentage, and settling time of 16.3 %, 31.1 %, and 64.9 %, respectively, as demonstrated by Table 3.

Instead, the second control technique, FLC, removes each of the oscillations and overshoots, leading to a comparatively prolonged rising time, while the FLC membership function tuned by PSO provides the most responsive control action, with rise time and settling time of 0.4 s and 0.7 s, respectively, leading to the most precise trajectory tracking capabilities (Supplementary Video 1, Supplementary Legends).

In future research, a nonlinear control algorithm will be developed that utilizes a disturbance observer to handle both external disturbances and model uncertainty, which are common challenges in controlling complex robotic systems. Experiments will be conducted using a physical CR as a validation platform. The experimental setup will include sensors and actuators that can measure and control the position and orientation of the CR with high accuracy and precision.

Data availability

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Code availability

The presented simulation analysis conducted using MATLAB Simulink, where it is available from the corresponding author, [Elsayed Atif Aner], upon reasonable request.

Received: 22 August 2023; Accepted: 21 December 2023

Published online: 06 January 2024

References

- Patidar, V. & Tiwari, R. Survey of robotic arm and parameters. *Int. Conf. Comput. Commun. Inform.* <https://doi.org/10.1109/ICCCI.2016.7479938> (2016).
- Zhao, G. *et al.* A tandem robotic arm inverse kinematic solution based on an improved particle swarm algorithm. *Front. Bioeng. Biotechnol.* <https://doi.org/10.3389/FBIOE.2022.832829/FULL> (2022).
- Kolachalama, S., Lakshmanan, S. Continuum robots for manipulation applications: A survey. <https://www.hindawi.com/journals/jr/2020/4187048/> (Accessed 18 July 2023) (2020).
- Wen, L. *et al.* A bio-inspired soft robotic arm: Kinematic modeling and hydrodynamic experiments. *J. Bionic Eng.* **15**(2), 204–219. <https://doi.org/10.1007/s42235-018-0016-x> (2018).
- Laschi, C. *et al.* Soft robot arm inspired by the octopus. *Adv. Robot.* **26**(7), 709–727. <https://doi.org/10.1163/156855312X626343> (2012).
- Wang, C., Puranam, V. R., Misra, S. & Venkiteswaran, V. K. A snake-inspired multi-segmented magnetic soft robot towards medical applications. *IEEE Robot. Autom. Lett.* **7**(2), 5795. <https://doi.org/10.1109/LRA.2022.3160753> (2022).
- Zhao, Y., Song, X., Zhang, X. & Lu, X. A hyper-redundant elephant's trunk robot with an open structure: Design, kinematics, control and prototype. *Chin. J. Mech. Eng. (English Ed.)* <https://doi.org/10.1186/S10033-020-00509-4> (2020).
- S. Wu, Q. Ze, J. Dai, N. Udipi, G. H. Paulino, and R. Zhao, "Stretchable origami robotic arm with omnidirectional bending and twisting," *Proc Natl Acad Sci U S A*, vol. 118, no. 36, Sep. 2021, doi: <https://doi.org/10.1073/PNAS.2110023118>.
- Webster, R. J. & Jones, B. A. Design and kinematic modeling of constant curvature continuum robots: A review. *Int. J. Robot. Res.* **29**(13), 1661–1683. <https://doi.org/10.1177/0278364910368147> (2010).
- Dupont, P., Simaan, N., Choset, H., Rucker, C. Continuum robots for medical interventions. In *Proc. of the IEEE*. <https://ieeexplore.ieee.org/abstract/document/9707607/> (Accessed 18 July 2023) (2022).
- Thuruthel, T. G. *et al.* Learning closed loop kinematic controllers for continuum manipulators in unstructured environments. *Soft Robot.* <https://doi.org/10.1089/soro.2016.0051> (2017).
- Zhang, J. *et al.* A survey on design, actuation, modeling, and control of continuum robot. *Cyborg. Bionic Syst.* <https://doi.org/10.34133/2022/9754697> (2022).
- Lu, J. *et al.* Kinematic modeling of a class of n-tendon continuum manipulators. *Adv. Robot.* **34**(19), 1254–1271. <https://doi.org/10.1080/01691864.2020.1812427> (2020).
- He, B., Wang, Z., Li, Q., Xie, H. & Shen, R. An analytic method for the kinematics and dynamics of a multiple-backbone continuum robot. *Int. J. Adv. Robot. Syst.* <https://doi.org/10.5772/54051> (2013).
- Webster, R. J. & Rucker, D. C. Parsimonious evaluation of concentric-tube continuum robot equilibrium conformation. *IEEE Trans. Biomed. Eng.* <https://doi.org/10.1109/TBME.2009.2025135> (2009).

16. Della Santina, C., Bicchi, A. On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. <https://ieeexplore.ieee.org/abstract/document/8961972/> (Accessed 8 June 2023) (2020).
17. Grazioso, S., Di Gironimo, G. & Siciliano, B. A geometrically exact model for soft continuum robots: The finite element deformation space formulation. *Soft Robot.* **6**(6), 790–811. <https://doi.org/10.1089/SORO.2018.0047> (2019).
18. El-Khatib, M., El-Khatib, M.F., Maged, S.A. Low level position control for 4-DOF arm robot using fuzzy logic controller and 2-DOF PID controller. In *2021 International Mobile, Intelligent, and Ubiquitous Computing*. <https://ieeexplore.ieee.org/abstract/document/9447617/> (Accessed 7 August 2023) (2021).
19. Till, J., Aloï, V. & Rucker, C. Real-time dynamics of soft and continuum robots based on Cosserat rod models. *Int. J. Robot. Res.* **38**(6), 723–746. <https://doi.org/10.1177/0278364919842269> (2019).
20. Janabi-Sharifi, F., Jalali, A. Cosserat rod-based dynamic modeling of tendon-driven continuum robots: A tutorial. <https://ieeexplore.ieee.org/abstract/document/9420666/> (Accessed 9 June 2023) (2021).
21. Amouri, A., Mahfoudi, C. & Zaatri, A. Dynamic modeling of a spatial cable-driven continuum robot using Euler-Lagrange method. *Int. J. Eng. Technol. Innov.* **10**(1), 60–74. <https://doi.org/10.46604/ijeti.2020.4422> (2020).
22. Amouri, A. *et al.* Dynamic modeling of a class of continuum manipulators in fixed orientation. *J. Intell. Robot. Syst.* **91**(3–4), 413–424. <https://doi.org/10.1007/s10846-017-0734-z> (2018).
23. Malik, H., Iqbal, A., Joshi, P., Agrawal, S. & Bakhsh, F. *Metaheuristic and Evolutionary Computation: Algorithms and Applications* (Springer Singapore, 2021). <https://doi.org/10.1007/978-981-15-7571-6>.
24. Aner, E. A., Awad, M. I. & Shehata, O. M. Modeling and trajectory tracking control for a multi-section continuum manipulator. *J. Intell. Robot. Syst. Theory Appl.* <https://doi.org/10.1007/S10846-023-01896-1> (2023).
25. Moussa, O., Mira, M., A. F. Behavioral assessment of various control laws formulations for position tracking of multi-sectioning modeled continuum robots. <https://ieeexplore.ieee.org/abstract/document/8988780/> (Accessed 9 June 2023) (2019).

Author contributions

E.A.A. conceptualized the proposed system and is the author of the proposed controller design and control architecture, develop and perform the simulation, program the designed application. Data analyses were performed by Professor M.I.A. and Dr. O.M.S. The first draft of the manuscript was written by E.A.A., where Professor M.I.A. and Dr. O.M.S. analyzes and revises previous versions of the manuscript. All authors read and approved the final manuscript.

Funding

Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-023-50551-0>.

Correspondence and requests for materials should be addressed to E.A.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024