ISSN 1392-1207. MECHANIKA. Vol. 31, No. 6, 2025: 567–577

Optimized Robust Fuzzy PID Controller for Suspended Cable-Driven Parallel Robots via Dragonfly Algorithm

Yasser H. ALWAN*,, Ahmed A. OGLAH*, Muayad S. CROOCK*****
*College of Control and Systems Engineering, University of Technology, Baghdad 10066, Iraq
**ITRDC, University of Kufa, Najaf, Iraq, E-mail: yasser.alwan@uokufa.edu.iq (Corresponding Author)
***College of Electrical Engineering, University of Technology, Baghdad 10066, Iraq
https://doi.org/10.5755/j02.mech.41438

1. Introduction

Cable-driven parallel robots (CDPRs) are a special type of parallel robot that uses cables instead of joints. CDPRs have several advantages over traditional rigid-link parallel robots, including a larger workspace, a higher payload-to-weight ratio, and lower manufacturing cost. The end-effector of CDPRs is connected by several flexible cables, which take the place of rigid links in traditional parallel robots. CDPRs can be classified into three groups based on the relationship between the number of cables and the degree of freedom (DOF): under-constrained, completely constrained, and redundantly constrained. They can also be classified into two types based on the dimension of the workspace: planar and spatial. The end-effector in a planar CDPR moves in a plane and has two or three DOFs, whereas the end-effector in a spatial CDPR operates in three dimensions and has more than three DOFs. The end-effector in spatial CDPRs can be suspended or non-suspended, depending on the location of the end-effector relative to the driving cables. If all driving cables are located above the end-effector position and gravity acts as an imaginary vertical cable to maintain balance, then the end-effector is called suspended. If at least one driving cable is below the end-effector position, then the end-effector is called non-suspended. Tension in cables of opposite directions in the non-suspended type can provide the action of push and pull to the end-effector. Given that gravity in the suspended type is a passive force, controlling a suspended CDPR is more challenging than controlling a non-suspended one [1], [2], [3], [4].

Suspended CDPRs have several advantages over non-suspended ones. The cables do not affect the workspace of the end-effector, and the payload is evenly distributed across the cables, resulting in high load capacity. Therefore, cable camera systems and lifting and moving loads are widely used applications in suspended CDPRs. On the other hand, they have some disadvantages, such as low rigidity in the vertical direction and susceptibility to instability under variable external disturbances. Suspended CDPRs can work with different DOFs depending on the structure design and the number of cables attached to the end-effector. In this paper, a 3D translational (3T) suspended CDPR is considered. In this type, the end-effector is a point mass without rotational motion. The position of the end-effector is where the cables' tension vectors intersect (usually at the center of mass of the end-effector) [1], [4].

Because of the proneness of suspended CDPRs to external disturbances that affect their stability, robust control is necessary when dealing with such CDPRs. Several studies have dealt with this problem, and a sliding mode controller (SMC) has been used in combination with other methods. SMC is a type of variable structure controller that has been proven to be robust to perturbations and external disturbances. In [5], a pick-and-place suspended CDPR was controlled using an adaptive control scheme via terminal sliding mode. An adaptive fuzzy system coupled with sliding mode was used in [6] and [7] to analyze suspended CDPR applications. Exponential sliding mode [8], dynamic sliding mode [9], adaptive fast terminal sliding mode [3], linear algorithms, and sliding mode control [10], have been studied and shown stability and acceptable performance when applied to suspended CDPRs. In [11], an optimized fuzzy proportional–integral–derivative (PID) control strategy was used in a suspended CDPR, in which a PID controller was optimized by fuzzy logic and an adaptive whale optimization algorithm. While these approaches demonstrated robustness and performance, the potential for unexplored control schemes to yield enhanced results warrants further investigation. Our previous work [12] on controlling a suspended CDPR using an adaptive fuzzy synergetic controller that was optimized using Dragonfly Algorithm (DA) produced promising robustness and accuracy results. The controller outperformed the one used in [7] on the same system. Although the real-time application requirements were satisfied, the algorithm was relatively demanding compared with simpler control schemes, motivating the need for a more computationally efficient design.

This work presents a new approach for 3T suspended CDPRs, in which the classical SMC is incorporated with an optimized fuzzy PID controller to achieve a combination of robustness and performance. In this approach, the equivalent control input is replaced with an optimized fuzzy PID controller. This method makes the controller minimally complicated and useful when the exact structure or parameters of the robot are uncertain. Reducing complexity is favorable if the controller is hybrid. DA, as a sort of swarm optimization algorithm that is suitable for engineering problems and has good convergence and minimum parameter tuning [13], is used to optimize the parameters of fuzzy PID and SMC.

This paper is structured as follows: Sections 2 and 3 develop the kinematic and dynamic models of the 3T suspended CDPR system. Sections 4 and 5 introduce the design of fuzzy PID and SMC controllers. A general introduction to DA is presented in section 6. Section 7 presents the results and discussions of simulation studies evaluating the controller's performance.

2. Kinematic Modeling of a 3T Suspended CDPR

Manuscript The workspace field of the suspended CDPR is surrounded by four pillars, where the cable pulleys are fixed on top. Four actuators with attached winches are located at the base of each pillar. The cables are the medium for conducting tension from the actuators to the end-effector and are guided by winches and pulleys. Fig. 1 represents the Cartesian coordinate system. The origin, $\boldsymbol{O}(0, 0, 0)$, is located at one of the corners of the base. The location of the end-effector is denoted by $\boldsymbol{P}(x, y, z)$ The corners of the top plane ($\boldsymbol{A}_i(x_{Ai}, y_{Ai}, z_{Ai})$, $i = 1, 2, 3, 4$) are the locations of the pulleys, and they represent the anchors of the cables. λ_i, $i = 1, 2, 3, 4$ indicates the lengths of the cables, and d_j, $j = 1, 2, 3$ represents the dimensions of the frame.

Applying the method of loop closure for the i^{th} cable leads to the inverse geometric transformation [3], [14], as shown below.

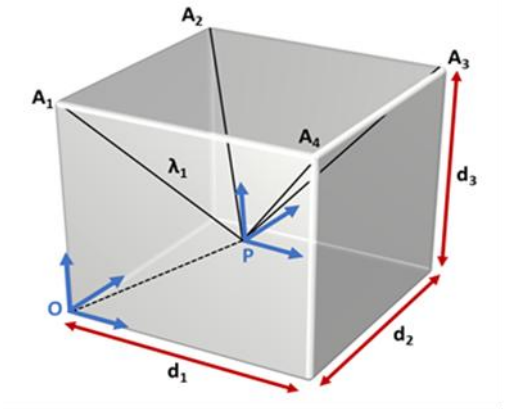


Fig. 1 Schematic of a suspended CDPR

$$\lambda_i \boldsymbol{u}_i = \boldsymbol{P} - \boldsymbol{A}_i, i = 1,2,3,4, \tag{1}$$

where λ_i, and $\boldsymbol{u}_i \in \mathbb{R}^4$ are the cable length and the unit vector of the i^{th} cable, respectively. $\boldsymbol{P} = [x\, y\, z]^T$, and $\boldsymbol{A}_i = [x_{Ai}, y_{Ai}, z_{Ai}]^T$. They are defined as follows:

$$\lambda_i = \| \boldsymbol{P} - \boldsymbol{A}_i \|, \boldsymbol{u}_i = \frac{\boldsymbol{P} - \boldsymbol{A}_i}{\| \boldsymbol{P} - \boldsymbol{A}_i \|}, i = 1, 2, 3, 4, \tag{2}$$

where $\|\bullet\|$ is the Euclidean norm. The time derivative of Eq. (1) gives the inverse kinematics of the CDPR.

$$\dot{\lambda}_i \boldsymbol{u}_i + \lambda_i \dot{\boldsymbol{u}}_i = \dot{\boldsymbol{P}}, i = 1, 2, 3, 4. \tag{3}$$

Multiplying both sides by $\boldsymbol{u}_i^T$ yields

$$\dot{\lambda}_i \boldsymbol{u}_i \boldsymbol{u}_i^T + \lambda_i \dot{\boldsymbol{u}}_i \boldsymbol{u}_i^T = \dot{\boldsymbol{P}} \boldsymbol{u}_i^T, i = 1, 2, 3, 4. \tag{4}$$

Because $\dot{\boldsymbol{u}}_i \boldsymbol{u}_i^T$ is zero, Eq. (4) becomes

$$\dot{\lambda} = \boldsymbol{u}_i^T \dot{\boldsymbol{P}}, i = 1, 2, 3, 4 \tag{5}$$

or

$$\dot{\boldsymbol{\Lambda}} = \boldsymbol{J}\dot{\boldsymbol{P}}, \tag{6}$$

where $\dot{\boldsymbol{\Lambda}} \in \mathbb{R}^4$ and $\boldsymbol{J} \in \mathbb{R}^{4\times3}$. $\boldsymbol{J}$ is expressed as:

$$\boldsymbol{J} = \begin{bmatrix} \frac{x - x_{Ai}}{\lambda_i} & \frac{y - y_{Ai}}{\lambda_i} & \frac{z - z_{Ai}}{\lambda_i} \end{bmatrix}, i = 1, 2, 3, 4. \tag{7}$$

That is the Jacobian matrix of the system.

3. Dynamic Modeling of a 3T Suspended CDPR

A dynamic mathematical model is crucial for designing the controller because it exposes the system's behavior. The method employed here involves using Lagrangian mechanics to obtain the motion equation [15]. The system is treated as a rigid body for this purpose. The mass and dynamics of the cables are overlooked because they are minor compared with the end-effector's mass. Typically, the motion equation is derived on the basis of the joint coordinates (in this case, the cable length), but this approach requires calculating the forward kinematic equations, which is not a straightforward task for CDPRs unless the structure is simple, such as planar or translational CDPRs. The use of task coordinates is more practical. The end-effector potential and kinetic energies (κ and ρ, respectively) can be expressed as follows in Cartesian coordinates [6]:

$$\kappa = \frac{1}{2} \dot{\boldsymbol{P}}^T \boldsymbol{M} \dot{\boldsymbol{P}}, \tag{8}$$

$$\rho = -mgz. \tag{9}$$

The partial derivatives of the Lagrangian ($\mathcal{L} = \kappa - \rho$) with respect to $\boldsymbol{P}$ and their time derivatives $\dot{\boldsymbol{P}}$ yield to the dynamic terms that are governing the system:

$$\boldsymbol{\tau} = \frac{d}{dt}\frac{\mathcal{L}}{\dot{\boldsymbol{P}}} - \frac{\mathcal{L}}{\boldsymbol{P}} = \boldsymbol{M}\ddot{\boldsymbol{P}} + \boldsymbol{G}, \tag{10}$$

where $\boldsymbol{M} = m\boldsymbol{I}_{3\times3}, \boldsymbol{G} = [0 \quad 0 \quad -mg]^T$, m is the end point mass, g is the acceleration of gravity, and $\boldsymbol{\tau} \in \mathbb{R}^3$ is the generalized force vector in task space.

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}. \tag{11}$$

Given that dealing with the generalized forces in the joint space (represented by cable tensions) rather than those in the task space is easier, a transformation between the two coordinate frames is needed. The transformation stems from the fact that the change in virtual work is the same in both coordinates [16], i.e.,

$$\delta \boldsymbol{W} = \boldsymbol{\tau}^T d\boldsymbol{P} = \boldsymbol{T}^T d\boldsymbol{\Lambda}, \tag{12}$$

where $\boldsymbol{W}$ is the virtual work, and $\boldsymbol{T} \in \mathbb{R}^4$ is the tension vector in joint space. According to Eq. (6), the rates of change are:

$$d\boldsymbol{\Lambda} = \boldsymbol{J} d\boldsymbol{P}. \tag{13}$$

Taking Eq. (13) into account, Eq. (12) can be written as:

$$\boldsymbol{\tau}^T d\boldsymbol{P} = \boldsymbol{T}^T \boldsymbol{J} d\boldsymbol{P} \,. \tag{14}$$

so that the transformation from forces in the task space to the joint space is given by

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{T}. \tag{15}$$

Now, Eq. (10) can be written as

$$\boldsymbol{J}^T \boldsymbol{T} = \boldsymbol{M}\ddot{\boldsymbol{P}} + \boldsymbol{G} + \boldsymbol{d} \,, \tag{16}$$

where $\boldsymbol{d}$ represents all the disturbances and uncertainties in $\boldsymbol{\tau}$, and it is assumed to be bounded by a scalar value d, i.e., $\|\boldsymbol{d}\| \le d$.

4. Fuzzy PID Controller

This section introduces the structure of the fuzzy PID controller. PID controllers are the most used in industrial control processes owing to their straightforward design and accepted performance under various operating conditions. However, a significant drawback of PID controllers is their sensitivity to noise and measurement inaccuracies because they can intensify input signal variations, leading to instability or oscillations. This issue is particularly true for CDPRs because they use cables, which are less rigid than those in other robot types. One strategy to enhance the performance of a PID controller is to integrate it with fuzzy logic [17], [18]. Fuzzy logic controllers (FLCs) operate based on the expertise of professionals, which means that they might not need a well-defined control plant model with explicitly known parameters, devoid of nonlinearities and uncertainties. The general architecture of an FLC includes a fuzzifier component, an inference engine component that combines fuzzified inputs with IF–THEN rules to derive the firing strength for each rule, and a defuzzifier component, which is a fuzzy-to-crisp output converter [19], [20], [21].

Many types of fuzzy logic systems exist, defined by their choice of membership functions (MFs) and the type of inference system. In this work, Gaussian MFs were chosen for their simple design, ease of representation and optimization, and computational efficiency with small rule bases. These characteristics are essential for real-time systems, such as robotics [22]. For the same reason, the product inference system is used. For x_i, $i = 1, 2, \ldots, p$ inputs, y_j, $j = 1, 2, \ldots, q$ outputs, L rules, $\mu_{F_i^k}$ input MFs, and $\mu_{G_j^k}$ output MFs ($k = 1, 2, \ldots, L$), the j^{th} fuzzy output can be expressed as [7]:

$$y_j = \frac{\sum_{k=1}^{L} \mu_{G_j^k} \left(\prod_{i=1}^{p} \mu_{F_i^k}(x_i) \right)}{\sum_{k=1}^{L} \left(\prod_{i=1}^{p} \mu_{F_i^k}(x_i) \right)} \,. \tag{17}$$

A fuzzy PID controller is built by feeding the error, its integral, and its derivative as clear inputs to the FLC. These signals are normalized by PID gains, similar to a standard PID controller. However, a three-input FLC requires a 3D rule base, which can be challenging to create because of several reasons [23]:

1. When using the rate of change in error, a human expert hardly perceives the third dimension of information, making it difficult to establish control rules.

2. When the sum of error is used, its linguistic values are challenging to quantify because different plants require different integral gains and steady-state values of the sum of error.

3. A 3D rule base can become extremely complex as the number of quantization levels (i.e., the number of MFs for each variable) for each dimension increases, causing the number of control rules to increase cubically with the number of quantization levels. Instead of a typical three-input fuzzy PID, an alternative fuzzy PID controller is used; it has two inputs (PD and scaling by K_p and K_d), the output of fuzzy logic is summed with its scaled integration (I) by K_i and the final sum is scaled by the output gain K_o (see Fig. 2). The overall structure keeps the actuation of PID of the input signal, in addition to the fuzzy rules. Using input/output scaling helps keep the fuzzy MFs within the normalized range of (−1, 1).

The fuzzy PD controller alone is a widely used controller for its stability, though its performance, especially in tracking error, is not optimum. The incorporation of an integrative action improves the steady-state error but leads to an overshoot in plant output, a well-known problem related to the integrative action [24], [25].

For solving the mentioned problem, a fuzzy PID controller is needed with an optimal set of gains. For obtaining these sets of optimal parameters, metaheuristic optimization algorithms are known for their efficiency. DA is one of these optimization methods, which is known to be fast and reliable.

5. Robust Controller Design

CDPR systems, in general, inherently encompass uncertainties in their parameters and are vulnerable to external disturbance. If the dynamic system is $\boldsymbol{\tau} = \boldsymbol{J}^T\boldsymbol{T} = \boldsymbol{M}^*\ddot{\boldsymbol{P}} + \boldsymbol{G}^* + \boldsymbol{f} + \boldsymbol{\delta}$, where $\boldsymbol{\tau}$ is the generalized tension vector, $\boldsymbol{T}$ is the generalized tension vector in joint space, $\ddot{\boldsymbol{P}}$ is the end-effector acceleration vector, $\boldsymbol{M}^* = \boldsymbol{M} + \Delta\boldsymbol{M}$ is the mass matrix, $\boldsymbol{G}^* = \boldsymbol{G} + \Delta\boldsymbol{G}$ is the gravity vector, $\boldsymbol{f}$ is the viscous and Coulomb friction vector, and $\boldsymbol{\delta}$ is the tension external disturbance vector, then the dynamic system can be written as $\boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{P}} + \boldsymbol{G}^* + \boldsymbol{d}$, where $\boldsymbol{d} = \Delta\boldsymbol{M}\ddot{\boldsymbol{P}} + \Delta\boldsymbol{G}^* + \boldsymbol{f} + \boldsymbol{\delta}$ is the generalized vector of all uncertainties and disturbances [7]; $\boldsymbol{d}$ is assumed to be bounded ($\|\boldsymbol{d}\| \le d$). An optimized fuzzy PID controller can guarantee performance; however, with this range of uncertainties and disturbances, the next step is to improve the controller to be a robust controller. SMC is a widely used robust controller in robotics, and literature has shown its effectiveness in rejecting disturbances and eliminating the effect of uncertainties in parameters. Nonetheless, chattering is one of the side effects of using such a controller. Performance is an issue without using an optimal equivalent control action in addition to the switching control action.

The proposed controller deals with these issues by using a fuzzy PID controller as an equivalent control action. The proposed controller is a hybrid one where both the fuzzy PID controller and the SMC are working together to improve the performance and eliminate the disturbance. Instead of using the traditional method to design the SMC, the

proposed method does not need the parameters of the plant to compensate for them, i.e., a model-free design approach and a less complicated design. For ensuring the stability and robustness of the proposed controller, a Lyapunov-based proof is considered. The sliding mode surface is defined as follows:

$$\boldsymbol{\sigma} = \dot{\boldsymbol{e}} + \boldsymbol{ce}\,, \tag{18}$$

where $\boldsymbol{e} = \boldsymbol{P}_d - \boldsymbol{P}$, $\boldsymbol{c} \in \mathbb{R}^3$ is a constant positive definite matrix, and $\boldsymbol{P}_d$ is the desired position trajectory. The derivative of the sliding surface is obtained as:

$$\dot{\boldsymbol{\sigma}} = \ddot{\boldsymbol{e}} + \boldsymbol{c}\dot{\boldsymbol{e}} = \ddot{\boldsymbol{P}}_d - \boldsymbol{M}^{-1}\left(-\boldsymbol{G} + \boldsymbol{J}^T\boldsymbol{T} - \boldsymbol{d}\right) + \boldsymbol{c}\ddot{\boldsymbol{e}}\,. \tag{19}$$

The candidate Lyapunov's function is defined as a positive definite one as follows:

$$V = \frac{1}{2}\boldsymbol{\sigma}^T\boldsymbol{M}\boldsymbol{\sigma}\,, \tag{20}$$

then its time derivative becomes

$$\dot{V} = \boldsymbol{\sigma}^T\boldsymbol{M}\dot{\boldsymbol{\sigma}}\,, \tag{21}$$

$$\dot{V} = \boldsymbol{\sigma}^T\left(\boldsymbol{M}\ddot{\boldsymbol{P}}_d + \boldsymbol{G} - \boldsymbol{J}^T\boldsymbol{T} + \boldsymbol{d} + \boldsymbol{Mc}\dot{\boldsymbol{e}}\right). \tag{22}$$

Considering $\boldsymbol{T} = \left(\boldsymbol{J}^T\right)^{\dagger}\left(\boldsymbol{u}_{eq} + \boldsymbol{u}_s\right)$, where $\left(\boldsymbol{J}^T\right)^{\dagger}$ is the pseudoinverse of $\boldsymbol{J}^T$, $\boldsymbol{u}_{eq}$ is the equivalent control signal that keeps the system state on the sliding surface, and $\boldsymbol{u}_s = \boldsymbol{K}_s sign(\boldsymbol{\sigma})$ is the switching control signal that handles the uncertainties and ensures that the system state converges to the sliding surface, yielding:

$$\dot{V} = \boldsymbol{\sigma}^T\left(\boldsymbol{u}_d - \boldsymbol{u}_{eq} - \boldsymbol{u}_s + \boldsymbol{d} + \boldsymbol{Mc}\dot{\boldsymbol{e}}\right), \tag{23}$$

where $\boldsymbol{u}_d = \boldsymbol{M}\ddot{\boldsymbol{P}}_d + \boldsymbol{G}$ is the desired control signal. Knowing that $\boldsymbol{u}_{eq}$ is the control signal that makes $\dot{\boldsymbol{\sigma}}$ from Eq. (19) equals to zero, it can be introduced as $\boldsymbol{u}_{eq} = \boldsymbol{u}_d + \boldsymbol{Mc}\dot{\boldsymbol{e}}$.

The control action provided by the optimized Fuzzy PID controller ($\boldsymbol{u}_{fuzzy}$) is one component of the final control action $\boldsymbol{u}$, which also includes $\boldsymbol{u}_{eq}$ and $\boldsymbol{u}_s$. It is plausible to incorporate $\boldsymbol{u}_{fuzzy}$ within $\boldsymbol{u}_{eq}$ to make the overall controller less complicated. Since $\boldsymbol{u}_{fuzzy}$ can be considered as a close value to $\boldsymbol{u}_d$, that leads to $\|\boldsymbol{u}_d - \boldsymbol{u}_{fuzzy}\| \le \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is the maximum tracking error using the optimized fuzzy PID controller via DA. Now, $\boldsymbol{u}_{eq} = \boldsymbol{u}_{fuzzy} + \boldsymbol{Mc}\dot{\boldsymbol{e}}$ makes Eq. (23) to be:

$$\dot{V} \le \boldsymbol{\sigma}^T\left(\boldsymbol{\varepsilon} - \boldsymbol{K}_s sign(\boldsymbol{\sigma}) + \boldsymbol{d}\right), \tag{24}$$

$$\dot{V} \le \boldsymbol{\sigma}^T\left(-\boldsymbol{K}_s sign(\boldsymbol{\sigma}) + \tilde{\boldsymbol{d}}\right), \tag{25}$$

$$\dot{V} \le \|\boldsymbol{\sigma}\|\left(-\boldsymbol{K}_s + \|\tilde{\boldsymbol{d}}\|\right), \tag{26}$$

$$\boldsymbol{T} = \left(\boldsymbol{J}^T\right)^{\dagger}\left(\boldsymbol{u}_{fuzzy} + \boldsymbol{K}_s sign(\boldsymbol{\sigma}) + \boldsymbol{Mc}\dot{\boldsymbol{e}}\right). \tag{27}$$

Inequality (26) follows by applying standard norm inequalities to bound the terms in (25). The term $\tilde{\boldsymbol{d}}$ is defined as $\tilde{\boldsymbol{d}} = \boldsymbol{d} + \boldsymbol{\varepsilon}$.

Fig. 2 shows the block diagram of the proposed controller. As long as $\boldsymbol{K}_s > \|\tilde{\boldsymbol{d}}\|$, the system is stable according to (26). It should be noticed that these stability criteria are not restricted because the fuzzy PID controller can compensate for the limitations of the SMC to some extent.

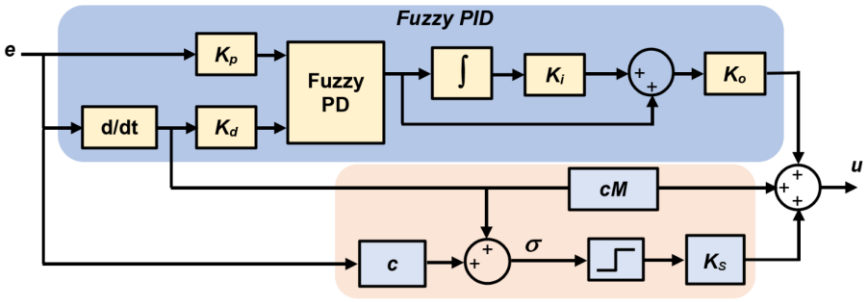


Fig. 2 Block diagram of Robust Fuzzy PID controller

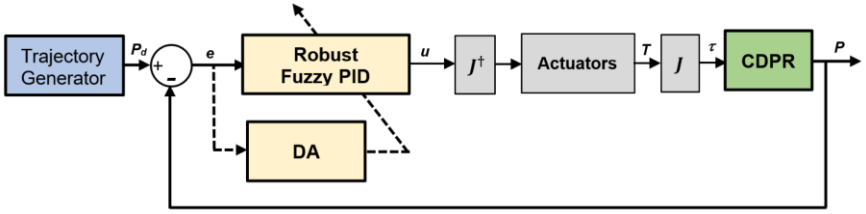


Fig. 3 Block diagram of the system with the proposed controller

First, DA is applied to optimize the gains of the fuzzy PID controller until $\boldsymbol{\varepsilon}$ is small enough not to affect the boundness of $\tilde{\boldsymbol{d}}$. Then, it is employed to optimize the value of $\boldsymbol{c}$ to ensure that the sliding mode is working properly, as shown in Fig. 3.

6. DA

DA is fundamentally inspired by the static and dynamic swarm behavior of dragonflies in nature. It models the social interactions of dragonflies during navigation, food

search, and enemy evasion, whether they are swarming statically or dynamically. DA emulates such static and dynamic swarm behavior, which closely resembles the two primary phases of metaheuristic optimization: exploration and exploitation. In the exploration phase, dragonflies form small subswarms and traverse various areas, akin to a static swarm. Conversely, in the exploitation phase, dragonflies swarm in large groups and move in a single direction, typical of a static swarm. These two phases are mathematically represented in the subsequent section [26].

DA has shown notable advantages in optimizing PID controllers [27], [28], particularly in achieving minimum fitness values and faster convergence rates when compared to algorithms like particle swarm optimization (PSO), differential evolution, biogeography-based optimization, teaching learning-based optimization, krill-herd algorithm, and grey wolf optimization [29]. These studies highlighted DA's ability to enhance dynamic response, reduce peak overshoot, and minimize settling time. Additionally, DA has proven effective in several other applications, showing significant improvement in execution times compared to PSO and honeybee behavior-inspired load balancing [30]. While DA has demonstrated strong capabilities in solving complex real-world problems, it also faces challenges like getting stuck in local optima due to its focus on exploitation more than exploration [31]. The "no free lunch" theorem further implies that no single algorithm can solve all optimization problems with the same effectiveness [32].

For mimicking the swarming behavior of dragonflies, five key concepts are employed: separation, alignment, cohesion, attraction to food sources, and distraction from enemies. These concepts enable the simulation of dragonfly behavior in dynamic and static swarms. DA is built upon the particle swarm optimization algorithm framework, and it primarily uses two vectors: the step vector and the position vector. These vectors record the direction/speed of movement and the position of the dragonflies. The primary equations for these vectors are as follows [26]:

$$\Delta\hat{\boldsymbol{X}}_{k+1} = \left(\hat{s}\hat{\boldsymbol{S}}_i + \hat{a}\hat{\boldsymbol{A}}_i + \hat{c}\hat{\boldsymbol{C}}_i + \hat{f}\hat{\boldsymbol{F}}_i + \hat{e}\hat{\boldsymbol{E}}_i\right) + \hat{\omega}\Delta\hat{\boldsymbol{X}}_k . \quad (28)$$

In this context, $\Delta\hat{\boldsymbol{X}}_k$ is the step vector of the position of the dragonfly at iteration k, and $\hat{\omega}$ is the inertia weight. $\hat{s}$, $\hat{a}$, $\hat{c}$, $\hat{f}$, and $\hat{e}$ represent the weights of separation, alignment, cohesion, food factor, and enemy factor, respectively. $\hat{\boldsymbol{S}}_i$, $\hat{\boldsymbol{A}}_i$, $\hat{\boldsymbol{C}}_i$, $\hat{\boldsymbol{F}}$, and $\hat{\boldsymbol{E}}_i$ denote the separation, alignment, cohesion, food source, and position of the enemy of the i^{th} individual, respectively. They are calculated as follows:

$$\hat{\boldsymbol{S}}_i = \sum_{j=1}^{N}\left(\hat{\boldsymbol{X}} - \hat{\boldsymbol{X}}_j\right), \quad (29)$$

$$\hat{\boldsymbol{A}}_i = \frac{1}{N}\sum_{j=1}^{N}\hat{\boldsymbol{X}}_j, \quad (30)$$

$$\hat{\boldsymbol{C}}_i = \frac{1}{N}\left(\sum_{j=1}^{N}\hat{\boldsymbol{X}}_j\right) - \hat{\boldsymbol{X}}, \quad (31)$$

$$\hat{\boldsymbol{F}}_i = \hat{\boldsymbol{X}}^{+} - \hat{\boldsymbol{X}}, \quad (32)$$

Initialize SuspendedCDPR with *Kinematic and Dynamic parameters*.
Phase 1
Initialize DA with *MaxIteration, PopulationSize*, and other *DA parameters*.
Define objective function as:
ITAE = ObjectiveFunction ($\mathbf{K}_p$, $\mathbf{K}_i$, $\mathbf{K}_d$, $\mathbf{K}_o$).
While k < *MaxIteration* **do**:
For each *agent* in *PopulationSize* **do**:
For each t in *MaxTime* **do**:
error = DesiredTrajectory(t) – System (SuspendedCDPR(t), FuzzyPID (t, $\mathbf{K}_p$, $\mathbf{K}_i$, $\mathbf{K}_d$, $\mathbf{K}_o$)).
ITAE = Function (t, **error**).
End For
Update FuzzyPID parameters $\mathbf{K}_p$, $\mathbf{K}_i$, $\mathbf{K}_d$, $\mathbf{K}_o$ for each *agent* using DA optimization.
End For
Select the best $\mathbf{K}_p$, $\mathbf{K}_i$, $\mathbf{K}_d$, $\mathbf{K}_o$ values using DA optimization.
End While

Phase 2
Reinitialize DA with *MaxIteration*, *PopulationSize*, and other *DA parameters*.
Define objective function as:
ITAE = ObjectiveFunction(**c**).
While k < *MaxIteration* **do**:
For each *agent* in *PopulationSize* **do**:
For each t in *MaxTime* **do**:
error = DesiredTrajectory(t) – System (SuspendedCDPR(t), optimized Fuzzy PID(t), SMC (t, $\mathbf{K}_s$, **c**)).
ITAE = Function (t, **error**).
End For
Update SMC parameters **c** for each *agent* using DA optimization.
End For
Select the best SMC parameter **c** based on DA optimization.
End While.

Fig. 4 Pseudocode for DA parameter optimization of the *robust fuzzy-PID* controller

$$\hat{\boldsymbol{E}}_i = \hat{\boldsymbol{X}}^{-} + \hat{\boldsymbol{X}}, \quad (33)$$

where $\hat{\boldsymbol{X}}$ is the position of the current dragonfly, $\hat{\boldsymbol{X}}_j$ indicates the position of the j^{th} neighboring solution, $\hat{\boldsymbol{X}}^{+}$ shows the position of a food source, $\hat{\boldsymbol{X}}^{-}$ is the position of an enemy, N is the number of neighboring dragonflies, and $\hat{\boldsymbol{X}}_j$ indicates the position of the j^{th} neighboring solution.

If an adjacent solution exists, the position vector is calculated as:

$$\hat{\boldsymbol{X}}_{k+1} = \hat{\boldsymbol{X}}_k + \Delta\hat{\boldsymbol{X}}_{k+1}. \quad (34)$$

Otherwise, a random walk is needed to explore a new area, as shown below:

$$\hat{\boldsymbol{X}}_{k+1} = \hat{\boldsymbol{X}}_k + \hat{Z}_s\hat{\boldsymbol{X}}_k \quad (35)$$

with

$$\hat{Z}_s = 0.01\left(\frac{a_1}{|a_2|^{\frac{1}{b}}}\right)\left(\frac{(s-1)!\cdot(1+b)\cdot sin\left(\frac{b\pi}{2}\right)}{(s-1)!\cdot b\cdot 2^{\left(\frac{b-1}{2}\right)}}\right)^{\frac{1}{b}}, \quad (36)$$

where a_1 and a_2 are random walk variables in the range [0, 1], b is a constant, and s is the dimension of $\hat{X}$

DA is used to optimize the 3×4 gains of the fuzzy PID controller, then the sliding surface parameter $\boldsymbol{c}$ of the total dimension of $\hat{X}$ equals to 12 and 3. The objective function is selected to minimize the integral time absolute error (ITAE) performance index [33]:

$$\text{ITAE} = \int_0^{\infty} t\left|e(t)\right| dt\,. \quad (37)$$

Using this index leads to controllers that maintain the robustness of the system, minimize the overshoot in the response, and exhibit high load disturbance rejection [34]. The pseudocode of the overall system, where two phases of optimization occur, is shown in Fig. 4.

7. Results and Discussion

The suspended CDPR system was modeled using MATLAB code. The kinematic and dynamic parameters of the CDPR were set as follows: mass of the end-effector m = 5 kg, acceleration of gravity g = 9.81 m/s^2 and dimensions of the frame $d_1 = d_2$ = 4 m and d_3 = 3 m. For demonstrating the effectiveness of the proposed controller, its performance was evaluated, and the corresponding results are illustrated. Then, the robustness was tested assuming a bounded disturbance and normal parameter uncertainty. The criteria that were utilized during the test were three error indices used in four tests.

The trajectory of the desired input to the system $P_d = \left[x_d\ y_d\ z_d\right]^T$ was chosen to be a circle following [6] and [7], as presented below:

$x_d = 0.8cos(0.1\pi t) + 1.8$,

$y_d = 0.8sin(0.1\pi t) + 2$,

$z_d = 1.5$.

The initial value of the end-effector position $\boldsymbol{P}_0 = \left[x_0\ y_0\ z_0\right]^T$ was chosen to be on the path of the trajectory as $[2.6, 2.0, 1.5]^T$. The initial end-effector velocity $\dot{\boldsymbol{P}}_0 = \left[\dot{x}_0\ \dot{y}_0\ \dot{z}_0\right]^T$ was set to $[0, 0, 0]^T$.

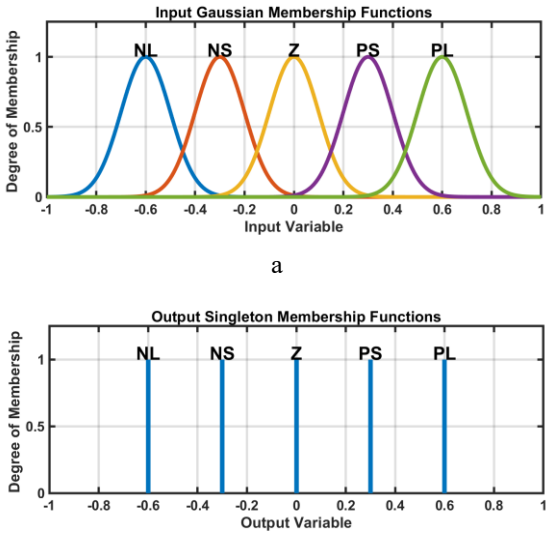
Initially, only the fuzzy PID controller was used. MFs were chosen to be five Gaussian functions (Fig.5, a) for each input and singleton functions for the output (Fig.5, b). The standard deviation of each MF was set to 0.1 and the mean vector to [−0.6, −0.3, 0, 0.3, 0.6]. This range is chosen to cover the span of [−1, 1] and scale the input to work within this interval. These MFs represent five fuzzy values: negative large (NL), negative small (NS), zero (Z), positive small (PS), and positive large (PL). The rule base is set as shown in Table 1 [35], [36]. When the error and its change are low, then the output is low, and when the error and its change are high, the output is high. The output increases faster with the increment of the change of error and slower with the increment of error.

For 250 iterations, DA, with a population size of 120 agents, was used to optimize 12 fuzzy PID gains for the best ITAE value. The number of agents was chosen to be 10 times the number of variables to add enough diversity in each swarm of dragonflies. To exchange between exploration and exploitation phases, the neighborhood radius r was expanded in proportion to the number of iterations. Additionally, adjusting the DA weight factors and the inertia factor ($\hat{s}, \hat{a}, \hat{c}, \hat{f}, \hat{e}$, and $\hat{\omega}$) during optimization can help achieve a balance between these two phases [26]. The upper limit of the gains is set to 100 by and the lower limit is set to just a small value above zero (to avoid zero and the result of overfitting by dropping one of the gains), by trial and error. The rest of the DA parameters were set as shown in Table 2 (case 1) following [37].

Table 1

Rule base of the PD fuzzy system

e/$\dot{e}$	NL	NS	Z	PS	PL
NL	NL	NL	NL	NS	Z
NS	NL	NL	NS	Z	PS
Z	NL	NS	Z	PS	PL
PS	NS	Z	PS	PL	PL
PL	Z	PS	PL	PL	PL



a

b

Fig. 5 MFs: a – Gaussian MFs used for fuzzification, b –Singleton MFs used at the output stage

The optimized gains were as follows:

$\boldsymbol{K}_p = diag\left[2.3813, 0.8393, 3.500\right]$,

$\boldsymbol{K}_i = diag\left[39.4577, 36.7564, 77.0781\right]$,

$\boldsymbol{K}_d = diag\left[1.5320, 1.750, 6.0350\right]$,

$\boldsymbol{K}_o = diag\left[80.7500, 88.0161, 79.0690\right]$.

With $\left\|\text{ITAE}\right\| = 0.1077$, the actual and desired positions of the end-effector are shown in Fig. 6.

Fig. 7 illustrates tracking errors in each axis. This shows the performance of the process when the system is disturbance-free.

A simulation of a disturbance and uncertainty signal was used to evaluate the robustness of the fuzzy PID controller. This signal is bounded and can be expressed as a vector, $\boldsymbol{d} = [4sin(10t), 2sin(10t), 4sin(10t)]^T$. This signal was added to the $\boldsymbol{\tau}$ vector. The tracking signal is illustrated

in Fig. 8. The system stability is maintained, but the performance is directly affected by the value of ***d***.

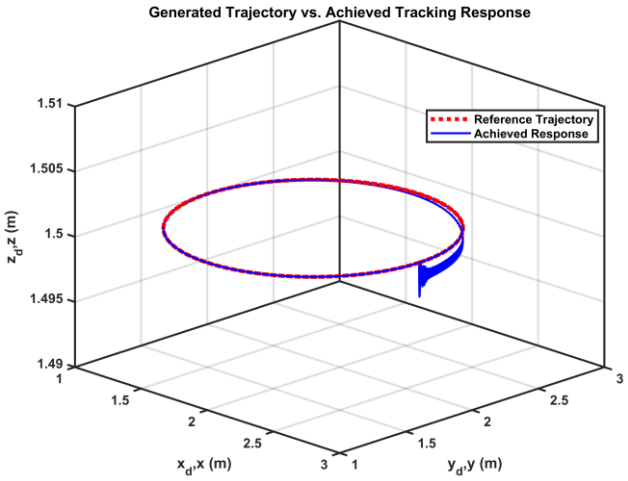


Fig. 6 Desired and actual positions of the end-effector when the *fuzzy PID* controller is applied

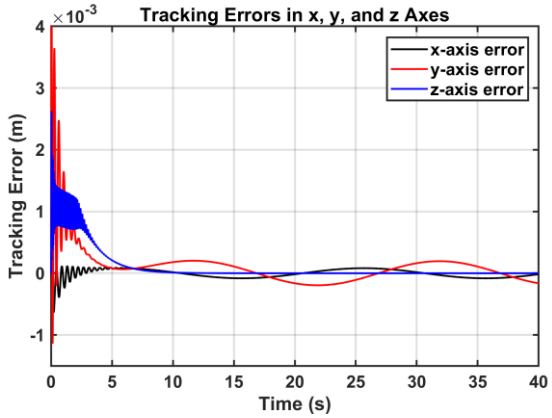


Fig. 7 Tracking errors in x,y,z axes when the fuzzy PID controller is applied

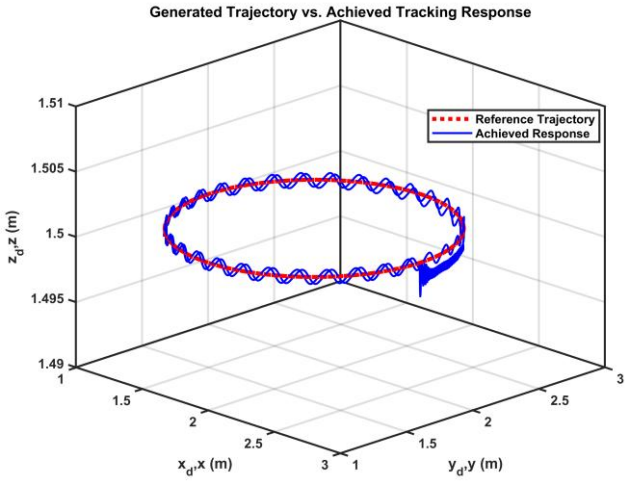


Fig. 8 Desired and actual positions of the end-effector when the fuzzy PID controller is applied, and the model of disturbance and uncertainties is added

Fig. 9 illustrates tracking errors in each axis. This shows the effect of the disturbance and uncertainties on the process.

Although ITAE is a good performance index in the optimization phase, it does not deliver the best depiction of

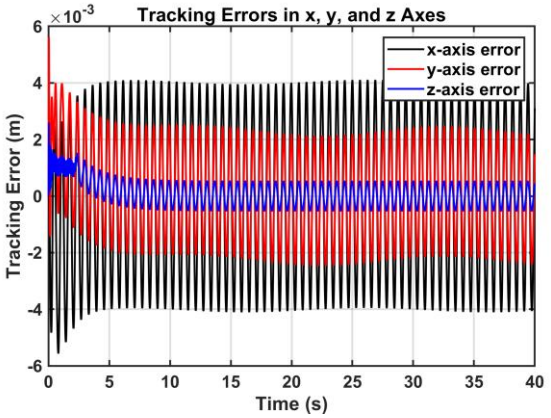


Fig. 9 Tracking errors in x, y, z axes when the fuzzy PID controller is applied, and the model of disturbance and uncertainties is added

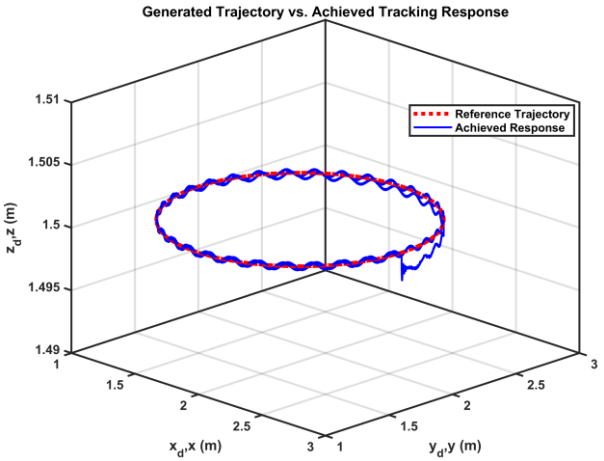


Fig. 10 Desired and actual positions of the end-effector when the robust fuzzy PID controller is applied, and the model of disturbance and uncertainties is added

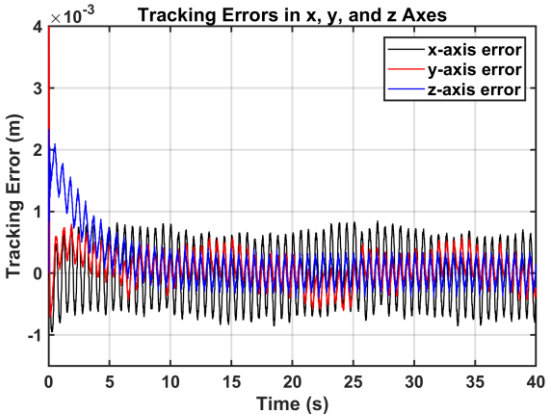
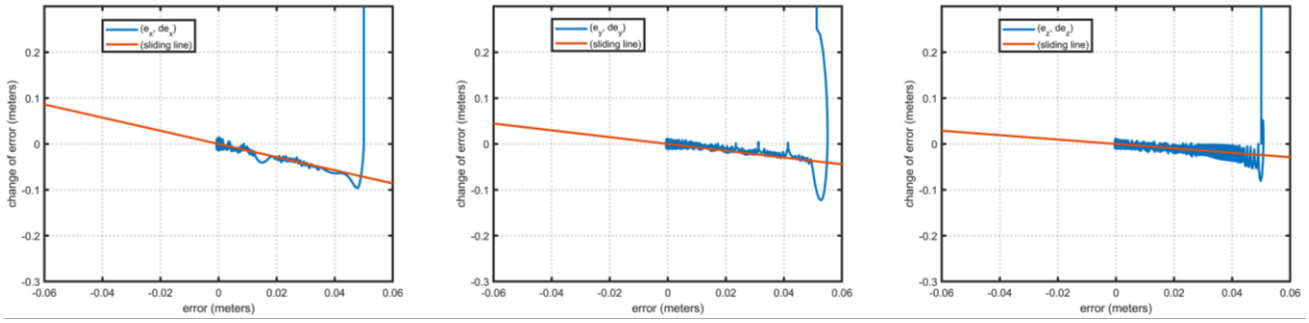


Fig. 11 Tracking errors in x,y,z axes when the robust fuzzy PID controller is applied, and the model of disturbance and uncertainties is added

error signal behavior. Two popular indices were used to show the error signal: root-mean-square error (RMSE) and maximum absolute error (MAE). They are expressed as [7]:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left|e(t)\right|^2} \quad \text{and} \quad \text{MAE} = \max_{t=1\to N}\left|e(t)\right|.$$

Table 3 compares the system responses integrating



a b c

Fig. 12 Phase planes of the reaching phase and sliding phase: a – errors/change of error of x-axis, b – errors/change of error of y-axis, c – errors/change of error of z-axis

Table 2

DA parameter setting when used to optimize the sliding surface parameter

Parameter	**Symbol**	**Value (case 1)**	**Value (case 2)**
Number of variables to be optimized	s	12	3
Lower bound of variables	lb	0.1	0.001
Upper bound of variables	ub	100	10
Nationhood hypersphere radius	r	$(ub-lb)\left(0.25+2\cdot\frac{k}{250}\right)$	$(ub-lb)\left(0.25+2\cdot\frac{k}{50}\right)$
Inertia weight	$\hat{\omega}$	$0.9+0.5\cdot\frac{k}{250}$	$0.9+0.5\cdot\frac{k}{50}$
Weight of separation	$\hat{s}$	$2\cdot\text{rand}\cdot\left(0.1+0.2\cdot\frac{k}{250}\right)$	$2\cdot\text{rand}\cdot\left(0.1+0.2\cdot\frac{k}{50}\right)$
Weight of alignment	$\hat{a}$	$2\cdot\text{rand}\cdot\left(0.1+0.2\cdot\frac{k}{250}\right)$	$2\cdot\text{rand}\cdot\left(0.1+0.2\cdot\frac{k}{50}\right)$
Weight of cohesion	$\hat{c}$	$2\cdot\text{rand}\cdot\left(0.1+0.2\cdot\frac{k}{250}\right)$	$2\cdot\text{rand}\cdot\left(0.1+0.2\cdot\frac{k}{50}\right)$
Weight of food factor	$\hat{f}$	$2\cdot\text{rand}$	$2\cdot\text{rand}$
Weight of enemy factor	$\hat{e}$	$0.1+0.2\cdot\frac{k}{250}$	$0.1+0.2\cdot\frac{k}{50}$
Random walk parameters	$[a_1, a_2, b]$	$[\text{rand}, \text{rand}, 1.5]$	$[\text{rand}, \text{rand}, 1.5]$

k is the iteration counter,
rand is a random number in [0,1)

Table 3

Error indices of the controllers

System	**ITAE**	**RMSE**	**MAE**
Fuzzy PID controller	0.1077	2.2711×10^{-4} m	0.0028 m
Fuzzy PID controller with added disturbance and uncertainty model	2.4148	0.0017 m	0.0043 m
Robust fuzzy PID controller	0.1085	1.7428×10^{-4} m	0.0025 m
Robust fuzzy PID controller with added disturbance and uncertainty model	0.3096	3.0974×10^{-4} m	0.0026 m

the optimized fuzzy PID controller without/with added disturbance and uncertainty model.

To add robustness to the system, this study used the proposed controller combining fuzzy PID and SMC. $\boldsymbol{K}_s$ values were chosen to be larger than the maximum bound of the added model of disturbance and uncertainties. The control signal was given by Eq. (27).

The values of the $\boldsymbol{c}$ vector play a crucial role in the controller performance; therefore, DA was used to optimize $\boldsymbol{c}$ values with ITAE as the performance index. DA was run for 50 iterations with a population size of 30 agents. The upper limit of the gains is set to 10, and the lower limit is set to just a small value above zero, by trial and error. The rest of the parameter settings are shown in Table 2 (case 2) following [37].

With $\|\text{ITAE}\| = 0.1085$, $\boldsymbol{c}$ was found to be:

$\boldsymbol{c} = diag[1.4356, 0.7418, 0.4811]$.

The values of $\boldsymbol{K}_s$ were critical for stability with the presence of disturbance and uncertainties. They should be large enough to exceed the value of $\|\boldsymbol{d}\|$ to guarantee stability, but not large enough to cause high chattering effects. The value of $\|\boldsymbol{d}\|$ is 6, Therefor, $\boldsymbol{K}_s$ were selected to be:

$\boldsymbol{K}_s = diag[7, 7, 7]$

Fig. 10 shows the tracking response of the proposed controller with added disturbance and uncertainty model. Clearly, it is less affected than the one with the fuzzy PID controller, while Fig. 11 illustrates tracking errors in each axis.

Table 3 shows that the proposed controller is better than the fuzzy PID controller alone in dealing with the rejection of disturbance and uncertainties. Similarly, the results are shown to be better than those of the adaptive fuzzy controller used in [7].

Their controller applied to the same system, with the same trajectory and disturbance signal. In their work, the RMSE was 8.9867×10^{-4} m, and the MAE was 2×10^{-2} m; by contrast, in the proposed controller, they were 3.0974×10^{-4} m and 2.6×10^{-3} m, respectively.

Fig. 12 shows the phase plane of the SMC within the proposed controller and the presence of disturbance and uncertainties. The initial values of $\boldsymbol{P}_0$ were set to $[2.55, 1.95, 1.45]^T$. The reason behind this choice of initial position is to show the reaching phase and sliding phase in a clear way in the graph.

While the adaptive fuzzy synergetic controller from [12] showed superior tracking performance (topping both the robust fuzzy PID controller proposed here and the adaptive fuzzy controller from [7]), this enhanced precision comes at a high computational cost. When tested on identical hardware, the run-time of the adaptive fuzzy synergetic approach was 11.186 s. In contrast, the proposed robust fuzzy PID controller achieves highly acceptable tracking performance with a significantly reduced computational load. Its run-time was found to be 1.184 s, representing a marked improvement of 89.41%. This substantial reduction in execution time is critical for real-time applications and demonstrates that the proposed scheme offers a better computationally efficient solution, making it highly practical for implementation on limited hardware resources without sacrificing stability and tracking quality.

8. Conclusions

A suspended CDPR is modeled and controlled using a robust fuzzy PID controller, which is an incorporation of fuzzy PID and SMC. Fuzzy PID combines the simple structure of a PID controller and the intelligence of an FLC. It is optimized using DA to be close enough to replace the equivalent control action of sliding mode control. Owing to the assumed uncertainties in the model parameters, this method is more practical than using conventional equivalent control. Moreover, DA is used to optimize the parameters of the sliding surface of SMC. The controller is tested mathematically and in simulation on disturbed and undisturbed suspended CDPRs. It realizes a balance between performance and robustness. The positive simulation results validate the controller's design and motivate the future work of implementing the controller on a physical CDPR for real-world experimental validation.

References

1. **Pott, A.** 2018. Cable-Driven Parallel Robots: Theory and Application. Berlin: Springer. 465p. https://doi.org/10.1007/978-3-319-76138-1.
2. **Hong, H.; Ali, J.; Ren, L.** 2018. A review on topological architecture and design methods of cable-driven mechanism, Advances in Mechanical Engineering 10(5): 1687814018774186. https://doi.org/10.1177/1687814018774186.
3. **Hosseini, M. I.; Harandi, M. J.; Seyedi, A. A. K.; Taghirad, H. R. D.** 2019. Adaptive Fast Terminal Sliding Mode Control of A Suspended Cable-Driven Robot, 27th Iranian Conference on Electrical Engineering (ICEE 2019): 985-990. https://doi.org/10.1109/IranianCEE.2019.8786501.
4. **Jomartov, A.; Tuleshov, A.; Kamal, A.; Abduraimov, A.** 2023. Simulation of suspended cable-driven parallel robot on SimulationX, International Journal of Advanced Robotic Systems 20(2): 17298806231161463. https://doi.org/10.1177/17298806231161463.
5. **El-Ghazaly, G.; Gouttefarde, M.; Creuze, V.** 2015. Adaptive Terminal Sliding Mode Control of a Redundantly-Actuated Cable-Driven Parallel Manipulator: CoGiRo. In: Pott, A.; Bruckmann, T. (Eds.), Cable-driven parallel robots, Mechanisms and Machine Science 32: 187–202. https://doi.org/10.1007/978-3-319-09489-2_13.
6. **Zi, B.** 2012. Fuzzy Control System Design and Analysis for Completely Restrained Cable-Driven Manipulators, In: Dadios, E. (Ed.), Fuzzy Logic - Controls, Concepts, Theories and Applications. InTech. https://doi.org/10.5772/35784.
7. **Liu, P.; Tian, H.; Cao, X.; Qiao, X.; Gong, L.; Duan, X.; Qiu, Y.; Su, Y.** 2022. Pick–and–Place Trajectory Planning and Robust Adaptive Fuzzy Tracking Control for Cable–Based Gangue–Sorting Robots with Model Uncertainties and External Disturbances, Machines 10(8): 714. https://doi.org/10.3390/machines10080714.
8. **Lv, W.; Tao, L.; Ji, Z.** 2017. Sliding Mode Control of Cable-Driven Redundancy Parallel Robot with 6 DOF Based on Cable-Length Sensor Feedback, Mathematical Problems in Engineering 2017: 1928673. https://doi.org/10.1155/2017/1928673.
9. **Khalilpour, S. A.; Khorrambakht, R.; Taghirad, H.; Cardou P.** 2019. Robust cascade control of a deployable cable-driven robot, Mechanical Systems and Signal Processing 127: 513-530. https://doi.org/10.1016/j.ymssp.2019.03.010.
10. **Picard, E.; Tahoumi, E.; Plestan, F.; Caro, S.; Claveau, F.** 2020. A new control scheme of cable-driven parallel robot balancing between sliding mode and linear feedback, IFAC-PapersOnLine 53(2): 9936-9943. https://doi.org/10.1016/j.ifacol.2020.12.2708.
11. **Zhou, B.; Wang, Y.; Zi, B.; Zhu, W.** 2024. Fuzzy Adaptive Whale Optimization Control Algorithm for Trajectory Tracking of a Cable-Driven Parallel Robot, IEEE Transactions on Automation Science and Engineering 21(4): 5149-5160. https://doi.org/10.1109/TASE.2023.3309049.
12. **Alwan, Y. H.; Oglah, A. A.; Croock, M. S.** 2025. Optimized Adaptive Fuzzy Synergetic Controller for Suspended Cable-Driven Parallel Robots, Automation 6(2): 15. https://doi.org/10.3390/automation6020015.
13. **Rahman, C. M.; Rashid, T. A.; Alsadoon, A.; Bacanin, N.; Fattah, P.; Mirjalili, S. A.** 2023. A survey on dragonfly algorithm and its applications in engineering, Evolutionary Intelligence 16: 1-21.

https://doi.org/10.1007/s12065-021-00659-x.
14. **Jomartov, A.; Tuleshov, A.; Kamal, A.; Abduraimov, A.** 2023. Design of a cable-driven parallel robot for landmine detection, SN Applied Sciences 5(11): 299. https://doi.org/10.1007/s42452-023-05533-2.
15. **Croock, M. S.; Abdullah, M. N.; Mousa, A. K.** 2015. Optimal Power Consumption Strategy for Smart Irrigation System Using Lagrange Multiplier, Sensors Letters 13(12): 1044-1049. https://doi.org/10.1166/sl.2015.3587.
16. **Lewis, F. L.; Dawson, D. M.; Abdallah, C. T.** 2004. Robot Manipulator Control: Theory and Practice, 2nd ed., Revised and Expanded. New York: Marcel Dekker. 614p.
17. **Borase, R. P.; Maghade, D. K.; Sondkar, S. Y.; Pawar, S. N.** 2021. A review of PID control, tuning methods and applications, International Journal of Dynamics and Control 9: 818–827. https://doi.org/10.1007/s40435-020-00665-4.
18. **Nguyen, N.-K.; Nguyen, D.-T.** 2021. A Comparative Study on PI – and PD – Type Fuzzy Logic Control Strategies, International Journal of Engineering Trends and Technology 69(7): 101-108. https://doi.org/10.14445/22315381/IJETT-V69I7P215.
19. **Fu, C.; Sarabakha, A.; Kayacan, E.; Wagner, C.; John, R.; Garibaldi, J. M.** 2018. Input Uncertainty Sensitivity Enhanced Nonsingleton Fuzzy Logic Controllers for Long-Term Navigation of Quadrotor UAVs, IEEE/ASME Transactions on Mechatronics 23(2): 725-734. https://doi.org/10.1109/TMECH.2018.2810947.
20. **Ghaleb, A. F.; Oglah, A. A.; Humaidi, A. J.; Al-Obaidi A.; Ibraheem, I.** 2023. Optimum of fractional order fuzzy logic controller with several evolutionary optimization algorithms for inverted pendulum, International Journal of Applied Science and Engineering 14(1): 1-12. https://doi.org/10.1556/1848.2021.00375.
21. **Mousa, A. K.; Croock, M. S.; Abdullah, M. N.** 2014. Fuzzy based Decision Support Model for Irrigation System Management, International Journal of Computer Applications 104(9): 14-20. https://doi.org/10.5120/18230-9177.
22. **Wu, D.** 2012. Twelve considerations in choosing between Gaussian and trapezoidal membership functions in interval type-2 fuzzy logic controllers, 2012 IEEE International Conference on Fuzzy Systems, Brisbane, QLD, Australia: 1-8. https://doi.org/10.1109/FUZZ-IEEE.2012.6251210.
23. **Yesil, E.; Güzelkaya, M.; Eksin, I**. 2003. The Third Triennial ETAI International Conference on Applied Automatic Systems, The Third Triennial ETAI International Conference on Applied Automatic Systems, Skopje, Macedonia: 105-112.
24. **Monteiro, J. R. B. A.; Pereira, W. C. A.; Santana, M. P.; Almeida, T. E. P.; Paula, G. T.; Santini, I.** 2013. Anti-windup method for fuzzy PD+I, PI and PID controllers applied in brushless DC motor speed control, 2013 Brazilian Power Electronics Conference: 865-871. https://doi.org/10.1109/COBEP.2013.6785216.
25. **Hasan, M. A.; Oglah, A. A.; Marie, M. J.** 2023. Optimal FOPI-FOPD controller design for rotary inverted pendulum system using grey wolves' optimization technique, Telkomnika 21(3): 657-666. https://doi.org/10.12928/telkomnika.v21i3.24383.
26. **Mirjalili, S. M.** 2016. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Computing and Applications 27: 1053-1073. https://doi.org/10.1007/s00521-015-1920-1.
27. **Simhadri, K.; Mohanty, B.; Rao, U. M.** 2019. Optimized 2DOF PID for AGC of Multi-area Power System Using Dragonfly Algorithm. In: Malik, H.; Srivastava, S.; Sood, Y.; Ahmad, A. (eds) Applications of Artificial Intelligence Techniques in Engineering. Advances in Intelligent Systems and Computing 698: 11-22. https://doi.org/10.1007/978-981-13-1819-1_2.
28. **Mishra, S.; Mohanty, B. K.** 2019. Step-back control of pressurized heavy water reactor by INFOPID using DA optimization, in: Malik, H.; Srivastava, S.; Sood, Y.; Ahmad, A. (eds) Applications of Artificial Intelligence Techniques in Engineering, Advances in Intelligent Systems and Computing 697: 497-507. https://doi.org/10.1007/978-981-13-1822-1_46.
29. **Guha, D.; Roy, P. K.; Banerjee, S.** 2018. Optimal tuning of 3 degree-of-freedom proportional-integral derivative controller for hybrid distributed power system using dragonfly algorithm, Computers and Electrical Engineering 72: 137-153. https://doi.org/10.1016/j.compeleceng.2018.09.003.
30. **Amini, Z.; Maeen, M.; Jahangir, M. R.** 2018. Providing a load balancing method based on dragonfly optimization algorithm for resource allocation in cloud computing, International Journal of Networked and Distributed Computing 6(1): 35-42. https://doi.org/10.2991/ijndc.2018.6.1.4.
31. **Alshinwan, M.; Abualigah, L.; Shehab, M.; Abd Elaziz, M.; Khasawneh, A. M.; Alabool, H.; Al Hamad, H.** 2021. Dragonfly algorithm: a comprehensive survey of its results, variants, and applications, Multimedia Tools and Applications 80(10): 14979-15016. https://doi.org/10.1007/s11042-020-10255-3.
32. **Meraihi, Y.; Ramdane-Cherif, A.; Acheli, D.; Mahseur, M.** 2020. Dragonfly algorithm: a comprehensive review and applications, Neural Computing and Applications 32(21): 16625-16646. https://doi.org/10.1007/s00521-020-04866-y.
33. **Kumar, I.; Kushwaha, K.** 2016. Optimization of automatic voltage regulator using genetic algorithm applying IAE, ITAE criteria, International Journal of Engineering Research and Applications 6(3, Part-6): 20-24.
34. **Awouda, A. A.; Mamat, R. B.** 2010. Refine PID tuning rule using ITAE criteria. In: 2nd International Conference on Computer and Automation Engineering (ICCAE): 171-176. https://doi.org/10.1109/ICCAE.2010.5451484.
35. **Kadhim, H. M.; Oglah, A. A.** 2020. Interval type-2 and type-1 fuzzy logic controllers for congestion avoidance in internet routers, IOP Conference Series: Materials Science and Engineering 881(1). https://doi.org/10.1088/1757-899X/881/1/012135.
36. **Raheema, R. S.; Hassan, M. Y.; Kadhim, S. K.** 2020. Simulation design of blood-pump intelligent controller based on PID-like fuzzy logic technique, Engineering and Technology Journal 38(A8): 1200-1213. https://doi.org/10.30684/etj.v38i8A.534.

37. **Mirjalili, S. M.** 2024. DA: Dragonfly Algorithm. MATLAB Central File Exchange. Available at: https://www.mathworks.com/matlabcentral/fileexchange/51035-da-dragonfly-algorithm (Accessed: October 30, 2024).

Y. H. Alwan, A. A. Oglah, M. S. Croock

OPTIMIZED ROBUST FUZZY PID CONTROLLER FOR SUSPENDED CABLE-DRIVEN PARALLEL ROBOTS VIA DRAGONFLY ALGORITHM

S u m m a r y

The kinematics and dynamics of a suspended cable-driven parallel robot with spatial of three degrees of freedom are presented. This robot has many applications, though, because of its structure, it is vulnerable to disturbance and uncertainties. This paper presents a proposed robust controller to deal with this issue. First, a fuzzy proportional–integral–derivative (PID) controller, which combines the merits of a fuzzy logic controller and the conventional widely used PID controller, is chosen as the main controller. Dragonfly algorithm (DA), as a relatively new metaheuristic algorithm, is known to have advantages over other classical ones. It is utilized to optimize PID gains until the required tracking error is achieved. Then, the chosen controller is incorporated with the robust classical sliding mode controller (SMC) in a way that balances the performance and robustness. DA is further exploited to optimize the parameters of SMC. Mathematical stability calculations and code modeling show the effectiveness of the proposed controller in performance and robustness.

Keywords: cable-driven parallel robot, suspended fuzzy logic controller, proportional–integral–derivative controller, dragonfly algorithm, sliding mode controller.

Received May 9, 2025
Accepted December 15, 2025