**Indonesian Journal of Electrical Engineering and Computer Science**
Vol. 33, No. 3, March 2024, pp. 1500~1510
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v33.i3.pp1500-1510

# Fractional-order PID controller tuned by particle swarm optimization algorithm for a planar CDPR control

**Hemama Aboud[1,2], Ammar Amouri[3], Abdelhakim Cherfia[3], Abdelaziz Mahmoud Bouchelaghem[1]**
[1]Department of Mechanical Engineering, Industrial Mechanics Laboratory, Badji Mokhtar Annaba University, Annaba, Algeria
[2]Department of Transport Engineering, Frères Mentouri Constantine 1 University, Constantine, Algeria
[3]Department of Mechanical Engineering, Laboratory of Mechanics, Frères Mentouri Constantine 1 University, Constantine, Algeria

**Article Info**

***Article history:***

Received Dec 4, 2023
Revised Jan 1, 2024
Accepted Jan 3, 2024

***Keywords:***

Cable-driven parallel robot
Dynamic modeling
Fractional-order PID controller
Optimization
Trajectory tracking

**ABSTRACT**

The use of cable-driven parallel robots (CDPRs) has been steadily increasing across various sectors due to their expansive workspaces, impressive payload-to-mass ratios, and cost-effective designs. Controlling these robots, particularly those with substantial actuation redundancy, can present challenges. This research paper proposes the implementation of a fractional-order proportional-integral-derivative (FOPID) controller to effectively regulate the end-effector of a planar CDPR with four actuation cables. The parameters of the controller are fine-tuned using the particle swarm optimization (PSO) algorithm to ensure optimal performance. The proposed controller's performance is evaluated through two numerical experiments: target tracking and trajectory tracking using a point-to-point approach. Furthermore, a comparative study is conducted to highlight the controller's performance, comparing the proposed FOPID controller with both the classical PID controller and an optimized PID controller. The achieved results demonstrate that the proposed controller exhibits superior performance in terms of tracking accuracy and smoothness of control signals when compared to the other controllers under investigation. As a result, the proposed controller design represents a substantial advancement in control performance and can be regarded as a promising control strategy for CDPRs.

***Corresponding Author:***

Hemama Aboud
Department of Mechanical Engineering, Industrial Mechanics Laboratory
Badji Mokhtar Annaba University
Annaba, Algeria
Email: aboudha088@gmail.com

## 1. INTRODUCTION

Cable-driven parallel robots (CDPRs) are a unique category of robotic mechanisms that stand out due to their reliance on cables for establishing a connection between a fixed base and a movable platform. With advantages surpassing other parallel robot designs, CDPRs offer a spacious workspace and substantial payload capacity by routing cables outside the workspace to support loads [1], [2]. Their heightened accuracy and precision are achieved through the manipulation of controlled cables with superior resolution. Moreover, CDPRs exhibit remarkable dexterity and flexibility, configuring cables in diverse arrangements tailored to specific tasks and environments [3], [4]. The applications of cable-driven parallel robots span various domains, including manufacturing [5], medical robotics [6], entertainment [7], and aerial robotics [8]. Excelling in tasks like pick-and-place operations [9], assembly [10], and inspection and monitoring tasks [11], [12], CDPRs prove to be versatile and efficient in a multitude of scenarios.

Nevertheless, challenges arise with cable-driven parallel robots concerning cable elasticity, precise calibration and maintenance, and the need for more intricate modeling and control compared to robots employing rigid-link mechanisms. To unlock the full potential of cable-driven parallel robots in diverse applications, the research paper commences by incorporating a more adaptable structure for the conventional PID controller. This novel approach, grounded in fractional calculus, is known as a fractional-order proportional-integral-derivative (FOPID) controller [13], [14].

The introduction of fractional calculus traces back to Leibniz, who first proposed the concept in a 1,684 letter to Tschirnhaus. This notion revolved around the differentiation and integration of functions with non-integer orders. Despite its initial introduction, fractional calculus required a substantial period for comprehensive study and development. In recent years, there has been a growing emphasis among researchers on topics related to fractional calculus. This increased focus has led to the identification of a diverse array of applications spanning multiple disciplines, such as physics [15], [16], engineering [17], [18], and neuroscience [19].

In the field of robotics, fractional calculus finds application in refining the control of robotic systems, leading to enhanced precision and efficiency in tasks like grasping, manipulation, and navigation. An illustration of such application is seen in the use of the conventional integer-order controller, such as the PID controller, in which a more advanced version called the FOPID has been developed [14]. The FOPID controller introduces additional parameters, specifically an integral-order parameter $\lambda$ and a differential-order parameter $\mu$, providing greater flexibility and capability in controlling complex systems compared to the classical PID controller. To meet the specified requirements for controller design, researchers employ a variety of algorithms to fine-tune parameters in FOPID controllers. These algorithms include bat optimization (BA), cuckoo search (CS), fruit fly optimization (FFO), particle swarm optimization (PSO), among others. However, certain researchers opt to utilize approximation techniques and toolboxes to streamline the implementation and practical application of fractional-order controllers. For additional details, readers are referred to [20]. Consequently, in the field of robotics, a diverse range of robots, such as parallel robots [21], [22], serial robot manipulators [23]−[25], cable-driven continuum robots [26], flexible-link robot manipulators [27], and mobile robots [28], [29], have been successfully controlled. Notably, the FOPID controller's application to cable-driven parallel robots has not been explored extensively. In an effort to address this gap, the present paper aims to employ the FOPID controller, coupled with the PSO algorithm, to precisely tune the parameters essential for effective control of the end-effector of a four-cable planar CDPR.

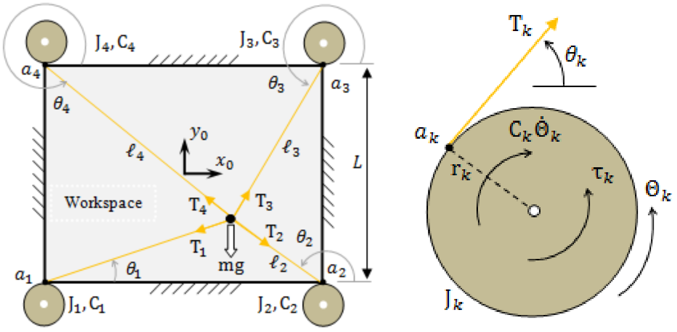
The structure of this paper unfolds as follows: In section 2, a comprehensive overview of the kinematics and dynamic modeling of a four-cable planar CDPR is presented. In section 3 delves into the development of the FOPID controller and outlines the process of tuning its parameters using PSO. Advancing to section 4, the simulation results are presented, including target tracking and trajectory tracking using the point-to-point technique, to demonstrate the efficacy and capabilities of the proposed controllers. Culminating the discussion, section 5 wraps up the present work and provides insights into potential avenues for future research.

## 2. MATHEMATICAL MODELING OF THE CDPR

### 2.1. Description and design of the CDPR

This paper concentrates on the investigation of a quadruple-cable planar CDPR, as depicted in Figure 1, where the schematic design is illustrated in Figure 1(a). The utilization of four flexible cables is a key aspect of this robotic system, allowing for the control of the end-effector's motion in a two-dimensional space. The robot's framework incorporates both a fixed platform and a mobile platform, intricately interconnected by a set of cables. These cables, securely affixed to the moving platform, are guided through pulleys to the stationary platform, forming a crucial part of the robot's mechanical design. To empower the robot in executing predetermined trajectories or fulfilling assigned tasks, the integration of motors becomes essential. Figure 1(b) illustrates a simplified schematic representation of a cable routing system commonly employed in CDPRs. The functionality of these motors is vital for propelling and regulating the motion of the cables.

Commencing our pursuit of the specified control and tracking objectives, we initiate by introducing the mathematical model that governs the dynamic behavior of the CDPR under examination. The crucial details of this model, including geometric parameters and symbols, are comprehensively outlined in Table 1. To provide additional clarity, some of these parameters are visually represented in Figure 1, offering a tangible connection between the mathematical framework and its corresponding geometric components.



(a) (b)

Figure 1. Quadruple-cable planar CDPR (a) schematic diagram of the planar four-cable CDPR and (b) schematic representation of the $k$ −th cable routing system

Table 1. Nomenclature of CDPR

| Symbols | Parameters |
|---|---|
| $a_k$ | Attachment points of the cable to the pulley |
| $C_k$ | Viscous damping coefficient for the shaft |
| $J_k$ | Inertia moment of the pulley |
| $L$ | Length of the CDPR's workspace side |
| $\ell_k$ | Cable length |
| m | Mass of the end-effector |
| $r_k$ | Pulley radius |
| $T_k$ | Cable tension |
| $\boldsymbol{X} = \{x, y\}^T$ | Generalized Cartesian coordinates |
| $\Theta_k$ | Pulley rotation angle |
| $\theta_k$ | Cable angle measured with respect to the $x$ −axis |
| $\tau_k$ | Motor torque |

### 2.2. Dynamic modeling

To formulate the equations of motion for the considered CDPR, a fundamental starting point is the calculation of dynamic models for both the end-effector and each of the actuation motors. Employing a two-dimensional state-space representation with generalized cartesian coordinates $X = \{x, y\}^T$, the end-effector's dynamic model for the CDPR is concisely articulated using Euler-Lagrange's method as (1):

$$\begin{pmatrix} \mathrm{m} & 0 \\ 0 & \mathrm{m} \end{pmatrix} \begin{Bmatrix} \ddot{x} \\ \ddot{y} \end{Bmatrix} - \begin{Bmatrix} 0 \\ \mathrm{mg} \end{Bmatrix} = \begin{Bmatrix} Q_x \\ Q_y \end{Bmatrix} \tag{1}$$

where m represents the mass of the end-effector, $\ddot{x}$ and $\ddot{y}$ denote the acceleration components of the end-effector in the $x$ −axis and $y$ −axis directions, respectively, and g is represents the acceleration constant. $Q_x$ and $Q_y$ signify the resultant force components of the four cable tensions in the $x$ −axis and $y$ −axis for the CDPR, respectively.

Ensuring equilibrium in the CDPR's end-effector requires acknowledging that the cumulative external forces from the cables ($T_k$, with $k = 1, \dots, 4$) must precisely balance the resultant external force $Q = \{Q_x, Q_y\}^T$. This critical relationship is succinctly formalized through (2):

$$Q = W(X)T \tag{2}$$

where the matrix $W(X)$ serves as a linkage, delineating the intricate connection between cable tension and resultant external forces. The expression for this matrix is provided as (3):

$$W(X) = -\begin{bmatrix} c(\theta_1) & c(\theta_2) & c(\theta_3) & c(\theta_4) \\ s(\theta_1) & s(\theta_2) & s(\theta_3) & s(\theta_4) \end{bmatrix} \tag{3}$$

where $c(.)$ and $s(.)$ denote $\cos(.)$ and $\sin(.)$, respectively. $\theta_k$, with $k = 1, \dots, 4$, represents the cable angle of cable $k$ measured with respect to the $x$ −axis (see Figure 1). This angle can be represented as a function of the generalized coordinate, as (4):

$$\theta_k = \tan^{-1}\left(\frac{y-a_{k,y}}{x-a_{k,x}}\right) \tag{4}$$

where $a_{k,x}$ and $a_{k,y}$ denote the attachment point cartesian coordinate of the cable to the pulley.

The dynamic behavior of each actuation motor can be defined according to [30], as in (5):

$$\mathrm{J}_k\ddot{\Theta}_k + \mathrm{C}_k\dot{\Theta}_k = \tau_k - \mathrm{r}_k\mathrm{T}_k \tag{5}$$

where, for each actuation motor $k$, $\Theta_k$ represents the rotation angle of the pulley, $\mathrm{J}_k$ is the rotational inertia associated with both the rotor and the motor's pulley, and $\mathrm{C}_k$ denotes the rotational viscous damping coefficient of the motor shaft. It is crucial to highlight that the model in this study exclusively examines the motion of the motors in isolation, disregarding external disturbances. This simplified model provides a foundational basis for analyzing the response characteristics of the motors.

Due to the constraint that cables can exert tensions only in the positive direction, signifying their inability to push, the relationship between cable tension, motor torque, and angular motion can be expressed as in (6) [30]:

$$\mathrm{T}_k = Positive\left(\frac{1}{\mathrm{r}_k}\left(\tau_k - \mathrm{J}_k\ddot{\Theta}_k - C_k\dot{\Theta}_k\right)\right) \tag{6}$$

where the term "$Positive(.)$" indicates that we consider only the positive values of each vector component and set any originally negative components to zero. As a matter of simplicity, we refrain from using this symbol in the subsequent developments.

Assuming that the angles of the pulleys are zero when the terminal member's position is at the center of the polygon $\mathrm{X} = \{0, 0\}^T$, the correlation between the rotation angles of the pulleys and the changes in the cable lengths can be expressed by the following equation. This initial condition serves as a starting point for the pulley angles, streamlining the formulation of the relationship between pulley rotations and the corresponding variations in cable lengths.

$$\Theta_k(x,y) = \frac{1}{\mathrm{r}}\left(\ell_k^{init} - \ell_k^{act}\right) \tag{7}$$

where $\ell_k^{init}$ and $\ell_k^{act}$, with $k = 1, \ldots, 4$, represent the initial and actual length of cable $k$, respectively. These lengths can be mathematically expressed as (8), (9) (see Figure 1(a)).

$$\ell_k^{init} = \sqrt{a_{k,x}^2 + a_{k,y}^2} \tag{8}$$

$$\ell_k^{act} = \sqrt{\left(x - a_{k,x}\right)^2 + \left(y - a_{k,y}\right)^2} \tag{9}$$

In a broader context, when all four pulley radii are equal ($\mathrm{r}_k = \mathrm{r}$), the dynamic behavior of the four actuation motors can be formulated as:

$$\begin{aligned}\mathrm{T} &= \frac{1}{\mathrm{r}}\left(\tau - \mathrm{J}\ddot{\Theta} - \mathrm{C}\dot{\Theta}\right)\\ &= \frac{1}{\mathrm{r}}\left(\tau - \mathrm{J}\left(\frac{d}{dt}\frac{\partial\Theta}{\partial\mathrm{X}}\dot{\mathrm{X}} + \frac{\partial\Theta}{\partial\mathrm{X}}\ddot{\mathrm{X}}\right) - \mathrm{C}\frac{\partial\Theta}{\partial\mathrm{X}}\dot{\mathrm{X}}\right)\end{aligned} \tag{10}$$

such that:

$$\frac{\partial\boldsymbol{\Theta}}{\partial\mathbf{X}} = -\frac{1}{\mathrm{r}}\begin{bmatrix}\frac{x-a_{1,x}}{\ell_1^{act}} & \frac{y-a_{1,y}}{\ell_1^{act}}\\ \vdots & \vdots\\ \frac{x-a_{4,x}}{\ell_4^{act}} & \frac{y-a_{4,y}}{\ell_4^{act}}\end{bmatrix} \tag{11}$$

By substituting (2) and (10), along with the first and second derivatives of the rotation angles of the pulleys as given in (7), into (1) and simplifying, the resulting dynamic model of the CDPR can be succinctly (12):

$$\mathrm{M}(X)\ddot{\mathrm{X}} + \mathrm{N}\left(\mathrm{X},\dot{X}\right)\dot{X} + \mathrm{K} = \mathrm{W}(X)\tau \tag{12}$$

where the components of the matrices outlined in (11) can be computed as (13).

$$\begin{cases} \mathrm{M}(X) = \mathrm{mr} + \mathrm{W}(X)\mathrm{J}\frac{\partial \Theta}{\partial \mathrm{X}} \\ \mathrm{N}(\mathrm{X}, \dot{X}) = \mathrm{W}(X)\left(\mathrm{J}\frac{d}{dt}\frac{\partial \Theta}{\partial \mathrm{X}} + \mathrm{C}\frac{\partial \Theta}{\partial \mathrm{X}}\right) \\ \mathrm{K} = [0 \quad -\mathrm{mg}]^{\mathrm{T}} \end{cases} \tag{13}$$

To model and analyze the input-output dynamics of the considered CDPR, it is crucial to represent the dynamic model in state space. The fourth-order Runge-Kutta method is subsequently employed for numerical computation. Introducing state variables as defined in (14) allows the dynamic model from (12) to be expressed in a compact form, as shown in (15);

$$\begin{cases} S_1 = x(t)\,, \quad S_3 = y(t) \\ S_2 = \dot{x}(t)\,, \quad S_4 = \dot{y}(t) \end{cases} \tag{14}$$

$$\dot{S}(t) = f(S,t) + h(S,t)U(t) \tag{15}$$

where $S(t)$ represents the vector of state variables, while $f(S,t)$ and $h(S,t)$ as nonlinear functions, and $U(t)$ as the command vector.

## 3. CONTROLLER DESIGN

### 3.1. Fractional-order PID controller structure

The fractional-order PID controller, denoted as $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$, serves as a generalized version of the classical PID controller, replacing the ordinary integral and derivative components with fractional operators represented by $\lambda$ and $\mu$, respectively [14]. These parameters can assume any arbitrary real number, typically chosen from the range of 0 to 2. The inclusion of these parameters enhances the controller's flexibility, leading to improved performance and increased robustness for the controlled system. Figure 2 visually depicts the relationship between PID and FOPID controllers as influenced by the values of the two parameters $\lambda$ and $\mu$.

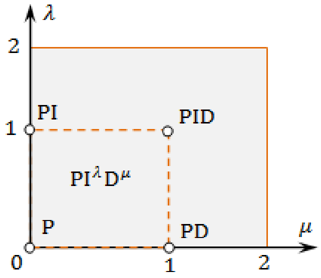


Figure 2. Relationships between conventional integer-order controllers and fractional-order controllers dependent on the two parameters $\lambda$ and $\mu$

In the context of the considered CDPR operating within a two-dimensional working space, the employment of two FOPID controllers is essential. One controller is specifically tailored for tracking the $x$ −input, while the other is dedicated to tracking the $y$ −input as shown in Figure 3. To exploit the geometrical symmetry of the cable-based robot, deliberately choosing similar coefficients for both controllers is critical. Hence, the expression of the control law in the time domain along the $x$ −axis and $y$ −axis can be succinctly stated as in (16):

$$\begin{cases} U_x(t) = K_{\mathrm{P}}e_x(t) + K_{\mathrm{I}}D^{-\lambda}e_x(t) + K_{\mathrm{D}}D^{\mu}e_x(t) \\ U_y(t) = K_{\mathrm{P}}e_y(t) + K_{\mathrm{I}}D^{-\lambda}e_y(t) + K_{\mathrm{D}}D^{\mu}e_y(t) \end{cases} \tag{16}$$

where the coefficients $K_{\mathrm{P}}, K_{\mathrm{I}}$ and $K_{\mathrm{D}}$ correspond to the proportional, integral, and derivative components, respectively. The control signals in the time domain for the $x$ −axis and $y$ −axis is denoted as $U_x(t)$ and

$U_y(t)$, respectively. The operator $D^{(.)}$ is a comprehensive operator that combines both integration and differentiation, commonly utilized in fractional calculus. The errors along the $x$ −axis and $y$ −axis is denoted as $e_x(t)$ and $e_y(t)$, respectively.

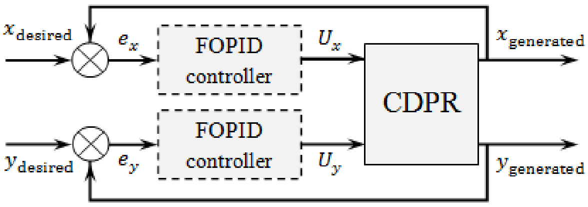


Figure 3. Schematic illustration of the utilized FOPID controllers for controlling the four-cable CDPR

### 3.2. PSO algorithm and parameters tuning

Within the realm of optimization algorithms presented in the literature for refining FOPID controllers, a notable approach is the PSO algorithm. First proposed by Kennedy and Eberhart in 1995 [31], this population-based search technique takes inspiration from the collective behavior observed in animals such as birds, bees, and schools of fish, collaborating when searching for food. The PSO algorithm's ability to mimic such cooperative behavior has proven effective in optimizing the parameters of FOPID controllers, contributing to their enhanced performance in various applications.

In the PSO algorithm, an individual solution to the optimization problem is designated as a particle, while the entire group of solutions is identified as a swarm. At the outset, particles are randomly distributed within the search space of the problem. Each particle, denoted as $p$, is defined by a position vector $\mathrm{x}_p^{iter}$ and a velocity vector $\mathrm{v}_p^{iter}$. The quality of these particles is assessed according to the value of the cost function. As the algorithm progresses, each particle remembers the best position it has reached, denoted as $pbest$, while the best position discovered by all particles within the swarm is referred to as $gbest$. The velocity and position of each particle $p$ between iterations $iter$ and $(iter + 1)$ are determined according to the following equations. Understanding the roles of individual particles, their personal best positions, and the swarm's overall best position is essential for comprehending how the PSO algorithm collectively converges towards optimal solutions as (17), (18):

$$\mathrm{v}_p^{iter+1} = w\mathrm{v}_p^{iter} + c_1\rho_1\left(pbest - \mathrm{x}_p^{iter}\right) + c_2\rho_2\left(gbest - \mathrm{x}_p^{iter}\right) \tag{17}$$

$$\mathrm{x}_p^{iter+1} = \mathrm{x}_p^{iter} + \mathrm{v}_p^{iter+1} \tag{18}$$

where $\mathrm{v}_p^{iter}$ represents the velocity of the particle, $\mathrm{x}_p^{iter}$ denotes the position of the particle, $w$ is the inertia weight, $c_1$ and $c_2$ are fixed constants, $\rho_1$ and $\rho_2$ are random numbers uniformly distributed in the range $[0, 1]$, $pbest$ signifies the local best position, and $gbest$ stands for the global best position. The inertia $w$ and the coefficients $c_1$ and $c_2$ influence the algorithm's behavior to achieve desired outcomes.

The PSO algorithm offers several advantages, including its straightforward implementation and the ability to achieve rapid convergence with minimal parameter tuning. While these strengths make PSO a popular choice, it is crucial to acknowledge that, as with many optimization algorithms, there is no guarantee of converging to the desired solution. To address the risk of the algorithm getting stuck in local minima, a population regeneration technique has been incorporated into the PSO algorithm [32], [33]. With these considerations in mind, the fundamental steps of the PSO algorithm utilized for fine-tuning the FOPID controller parameters are outlined in Algorithm 1. Subsequently, Algorithm 2 provides the structure for tuning FOPID controller parameters using the PSO algorithm.

Algorithm 1. Particle swarm optimization

```
Input: PSO parameters, number of iterations ($\max\_iter$).
Initialize: Randomly place particles and set their initial velocities within the search
space.
Set: $iter$ (iteration counter) to 0.
while the stopping criteria are not fulfilled, repeat the following steps:
   for each particle $p$ in the swarm, do:
      Evaluate the cost function for the current position.
   end for
```

```
    Select the global best position ($gbest$) from the entire swarm.
       for each particle $p$ in the swarm, do:
          Select the best position ($pbest$) for particle $p$ based on its own history.
          Update the position of particle $p$ based on Equations 13 and 14.
       end for
    Increment $iter$ by 1.
    if $iter$ reaches the maximum allowed iterations ($iter\_max$), then:
       Reinitialize the particles and their velocities randomly within the search space.
       Reset $iter$ to 0.
    end if
end while
Output: The best position found, $p$.
```

Algorithm 2. Tuning FOPID controller parameters with PSO

Here, the cost function to be minimized for tuning the FOPID controller parameters is defined as the square error between the two outputs along the x-axis and y-axis.

```
Input: PSO parameters, number of iterations ($\max\_iter$). Parameters range of FOPID
controller, convergence criteria.
Initialization: Randomly distribute the particles within the search space and
initialize them with their initial velocities.
for iteration = $1{:}\max\_iter$
  for particle = $1{:}swarm\_size$
     Update particle's velocity and position.
     Evaluate cost function for new position.
     Update the best position.
     Update the global best position.
  end
if convergence criteria is met.
  Break;
end
Output: Optimized FOPID parameters.
```

## 4. NUMERICAL EXPERIMENTS AND ANALYSIS

To assess the performance and effectiveness of the proposed FOPID controller, two simulation scenarios were conducted. The first simulation focuses on target tracking, investigating the impact of the two non-integer parameters, $\lambda$ and $\mu$. In this scenario, a comparative study was conducted to highlight the controller's performance, comparing it with both the classical PID controller and an optimized PID controller. The second scenario is dedicated to circular trajectory tracking. These simulations were performed using the MATLAB software. The essential parameters for the simulation of the robotic system are outlined in Table 2 [30], while the optimal controller parameters are provided in Table 3. In both simulation scenarios, a sampling time of 0.01 seconds was employed.

Table 2. CDPR parameters

| Parameter | $L$ | m | r | $J_k$ | $C_k$ | g |
|---|---|---|---|---|---|---|
| Value | 65.8 cm | 1 kg | 5 cm | 8 kg.cm$^2$ | 1 N.cm.s | 10 m/s$^2$ |

Table 3. Optimal FOPID, PID and OPID controller parameters

| Parameters | $K_P$ | $K_I$ | $K_D$ | $\lambda$ | $\mu$ |
|---|---|---|---|---|---|
| FOPID | 110 | 70 | 0.3 | 1.5 | 0.65 |
| PID | 110 | 70 | 0.3 | 1 | 1 |
| OPID | 70 | 40 | 2 | - | - |

### 4.1. Target tracking

To assess the effectiveness of the intoduced controller, the cartesian coordinates $[0.3, 0.3]$ and $[0.0, -0.4]$ (m), representing two desired points within the robot's workspace, are set as targets for the CDPR end-effector to reach. The tracking profiles are depicted in Figure 4, enabling a comparison with the classical PID controller and OPID controller. Figure 4 clearly illustrates the superior control performance of the FOPID controller compared to the PID and OPID controllers. Furthermore, to explore the influence of the FOPID parameters on control performance, specifically the integral-order $\lambda$ and the derivative-order $\mu$, Figures 5 and 6 present a comparison of the step response properties for the aforementioned target tracking $[0.3, 0.3]$ (m) as $\lambda$ and $\mu$ vary within the interval $[0.25: 0.25: 1.5]$.

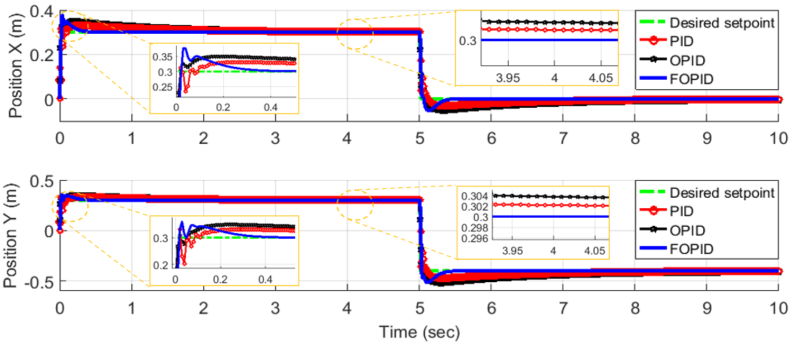


Figure 4. Comparison of target tracking responses among FOPID, PID, and OPID controllers

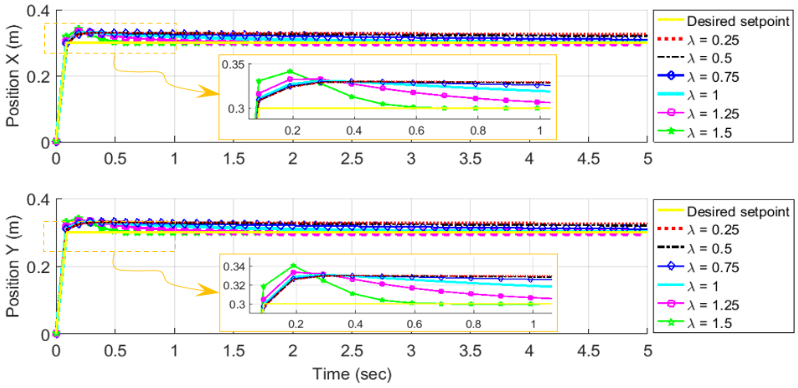


Figure 5. Step responses for $\mu = 1$ with varied $\lambda$ values

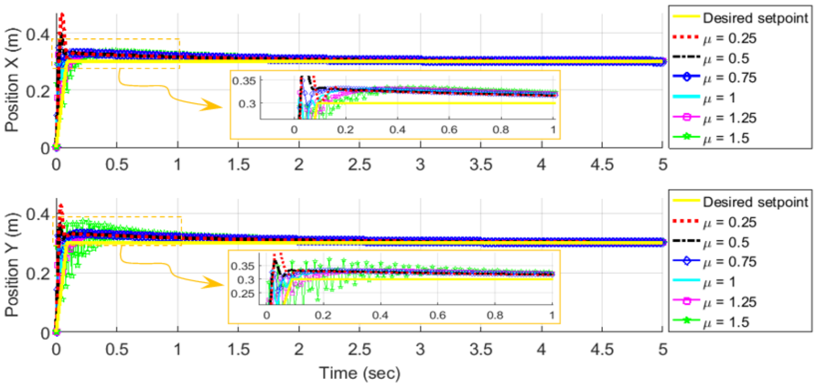


Figure 6. Step responses for $\lambda = 1$ with varied $\mu$ values

### 4.2. Trajectory tracking via point-to-point approach

In the second numerical experiment, utilizing the same parameters as mentioned above, a circular-shaped trajectory depicted in Figure 7 is employed to assess the effectiveness of the introduced controller. The curves of the desired and generated trajectory closely overlap in this figure, indicating the controller's capability to accurately track the circular trajectory. Furthermore, the necessary control signals for tracking the circular-shaped trajectory and the length variation in the cables are presented in Figures 8 and 9, respectively.

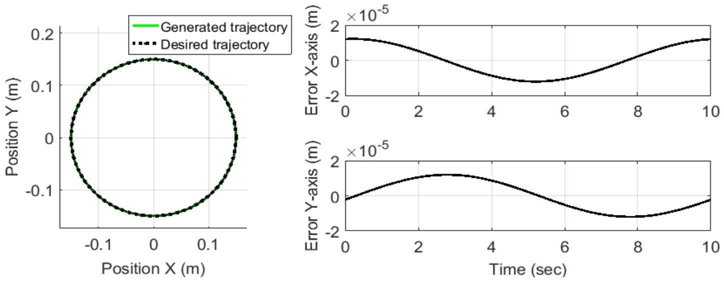


Figure 7. Desired cartesian trajectory and its generated counterpart, accompanied by the errors present among them

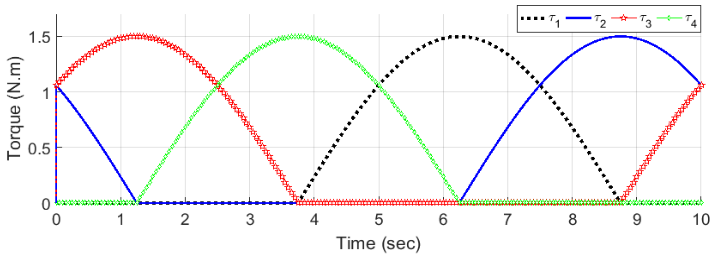


Figure 8. Control signals for tracking the circular-shaped trajectory

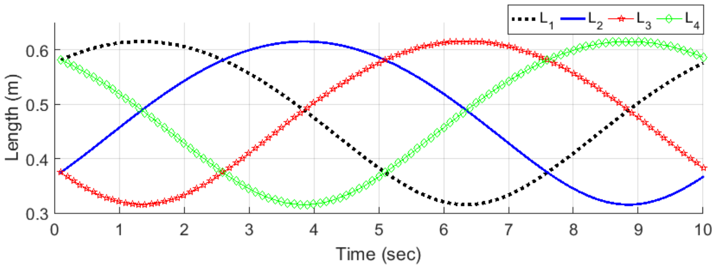


Figure 9. Cable length variation during tracking of the circular-shaped trajectory

In conclusion, the simulation results obtained underscore the superior capabilities of the introduced FOPID controller. Through a comprehensive comparative analysis involving classical PID and OPID controllers, it becomes evident that the FOPID excels in terms of both tracking accuracy and control signal smoothness. The enhanced performance can be attributed to the fractional order of the integral and derivative components, endowing the FOPID with greater flexibility and efficacy. The control signals produced by the introduced FOPID controller exhibit smoother characteristics and lower amplitude compared to both the PID and OPID controllers, indicating a substantial improvement in the precision and smoothness of the control signals. Furthermore, the FOPID controller achieves precise control of the studied planar CDPR end-effector, leading to positioning errors that are practically negligible. In summary, the FOPID controller emerges as a robust and effective choice for applications requiring precise control and minimal tracking errors in dynamic systems.

## 5. CONCLUSION

In this paper, we employ a FOPID controller tuned through the PSO algorithm to tackle the trajectory tracking challenge in a position control system specifically designed for a planar CDPR. The effectiveness of the introduced FOPID controller is assessed in target tracking and trajectory tracking via point-to-point technique scenarios, establishing a comparison with both the classical PID controller and the OPID controller. The analysis reveals that the introduced FOPID controller outperforms the PID and OPID controllers in both efficiency and accuracy. This superiority is particularly noteworthy in the context of trajectory tracking for the planar CDPR, emphasizing the potential of FOPID control for enhancing precision and performance in such robotic systems. As a forward-looking aspect of this research, our intent is to implement the proposed FOPID controller in the context of three-dimensional operation for CDPRs. Additionally, we plan to explore FOPID parameter tuning using alternative optimization algorithms, further contributing to the optimization and adaptability of the control strategy. This research aims to broaden the applicability and robustness of the FOPID controller in various operational scenarios, marking a stride in advancing control systems for cable-based parallel robots.

## REFERENCES

[1] S. Qian, B. Zi, W. W. Shang, and Q. S. Xu, "A review on cable-driven parallel robots," *Chinese Journal of Mechanical Engineering (English Edition)*, vol. 31, no. 4, p. 66, Dec. 2018, doi: 10.1186/s10033-018-0267-9.

[2] T. P. Tho and N. T. Thinh, "Using a cable-driven parallel robot with applications in 3D concrete printing," *Applied Sciences (Switzerland)*, vol. 11, no. 2, pp. 1–24, Jan. 2021, doi: 10.3390/app11020563.

[3] X. Tang, "An overview of the development for cable-driven parallel manipulator," *Advances in Mechanical Engineering*, vol. 2014, p. 823028, Jan. 2014, doi: 10.1155/2014/823028.

[4] A. Pott, *Cable-driven parallel robots: Theory and application*, vol. 120. Cham: Springer International Publishing, 2018.

[5] D. Gueners, H. Chanal, and B. C. Bouzgarrou, "Stiffness optimization of a cable driven parallel robot for additive manufacturing," in *Proceedings - IEEE International Conference on Robotics and Automation*, May 2020, pp. 843–849, doi: 10.1109/ICRA40945.2020.9197368.

[6] I. Hamida, M. A. Laribi, A. Mlika, L. Romdhane, S. Zeghloul, and G. Carbone, "Multi-Objective optimal design of a cable driven parallel robot for rehabilitation tasks," *Mechanism and Machine Theory*, vol. 156, 2021, doi: 10.1016/j.mechmachtheory.2020.104141.

[7] P. Miermeister *et al.*, "The CableRobot simulator large scale motion platform based on Cable Robot technology," in *IEEE International Conference on Intelligent Robots and Systems*, Oct. 2016, vol. 2016-Novem, pp. 3024–3029, doi: 10.1109/IROS.2016.7759468.

[8] Q. Jiang and V. Kumar, "Determination and stability analysis of equilibrium configurations of objects suspended from multiple aerial robots," *Journal of Mechanisms and Robotics*, vol. 4, no. 2, May 2012, doi: 10.1115/1.4005588.

[9] Z. Zhang, Z. Shao, L. Wang, and A. J. Shih, "Optimal design of a high-speed pick-and-place cable-driven parallel robot," in *Mechanisms and Machine Science*, vol. 53, 2018, pp. 340–352.

[10] K. Iturralde *et al.*, "Cable-driven parallel robot for curtain wall module installation," *Automation in Construction*, vol. 138, p. 104235, Jun. 2022, doi: 10.1016/j.autcon.2022.104235.

[11] M. Bartoš, V. Bulej, and I. Kuric, "Conceptual design and simulation of cable-driven parallel robot for inspection and monitoring tasks," *MATEC Web of Conferences*, vol. 357, p. 02024, Jun. 2022, doi: 10.1051/matecconf/202235702024.

[12] S. Seriani, P. Gallina, and A. Wedler, "A modular cable robot for inspection and light manipulation on celestial bodies," *Acta Astronautica*, vol. 123, pp. 145–153, Jun. 2016, doi: 10.1016/j.actaastro.2016.03.020.

[13] G. A. Sultan, A. F. Sheet, S. M. Ibrahim, and Z. K. Farej, "Speed control of DC motor using fractional order PID controller based on particle swarm optimization," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 3, pp. 1345–1353, Jun. 2021, doi: 10.11591/ijeecs.v22.i3.pp1345-1353.

[14] I. Podlubny, "Fractional-order systems and PIλDμ-controllers," *IEEE Transactions on Automatic Control*, vol. 44, no. 1, pp. 208–214, Jan. 1999, doi: 10.1109/9.739144.

[15] Z. Wang, X. Wang, Y. Li, and X. Huang, "Stability and hopf bifurcation of fractional-order complex-valued single neuron model with time delay," *International Journal of Bifurcation and Chaos*, vol. 27, no. 13, p. 1750209, Dec. 2017, doi: 10.1142/S0218127417502091.

[16] D. Liu, S. Zhao, X. Luo, and Y. Yuan, "Synchronization for fractional-order extended Hindmarsh-Rose neuronal models with magneto-acoustical stimulation input," *Chaos, Solitons and Fractals*, vol. 144, 2021, doi: 10.1016/j.chaos.2020.110635.

[17] R. Mohamed, M. Helaimi, R. Taleb, H. A. Gabbar, and A. M. Othman, "Frequency control of microgrid system based renewable generation using fractional PID controller," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 2, pp. 745–755, Aug. 2020, doi: 10.11591/ijeecs.v19.i2.pp745-755.

[18] L. H. Abood and B. K. Oleiwi, "Design of fractional order PID controller for AVR system using whale optimization algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 3, pp. 1410–1418, Sep. 2021, doi: 10.11591/ijeecs.v23.i3.pp1410-1418.

[19] O. Brandibur and E. Kaslik, "Stability properties of a two-dimensional system involving one Caputo derivative and applications to the investigation of a fractional-order Morris–Lecar neuronal model," *Nonlinear Dynamics*, vol. 90, no. 4, pp. 2371–2386, Dec. 2017, doi: 10.1007/s11071-017-3809-2.

[20] K. Bingi, R. Ibrahim, M. N. Karsiti, S. M. Hassan, and V. R. Harindran, *Scilab based toolbox for fractional-order systems and PID controllers*, vol. 264. Cham: Springer International Publishing, 2020.

[21] A. Al-Mayyahi, A. A. Aldair, and C. Chatwin, "Control of a 3-RRR planar parallel robot using fractional order PID controller," *International Journal of Automation and Computing*, vol. 17, no. 6, pp. 822–836, Dec. 2020, doi: 10.1007/s11633-020-1249-9.

[22] L. Angel and J. Viola, "Fractional order PID for tracking control of a parallel robotic manipulator type delta," *ISA Transactions*, vol. 79, pp. 172–188, Aug. 2018, doi: 10.1016/j.isatra.2018.04.010.

[23] M. A. Faraj and A. M. Abbood, "Fractional order PID controller tuned by bat algorithm for robot trajectory control," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, pp. 74–83, 2021, doi: 10.11591/ijeecs.v21.i1.pp74-83.

[24] L. Angel and J. Viola, "Control performance assessment of fractional-order PID controllers applied to tracking trajectory control of robotic systems," *WSEAS Transactions on Systems and Control*, vol. 17, pp. 62–73, Feb. 2022, doi: 10.37394/23203.2022.17.8.

[25] R. Sharma, P. Gaur, and A. P. Mittal, "Performance analysis of two-degree of freedom fractional order PID controllers for robotic manipulator with payload," *ISA Transactions*, vol. 58, pp. 279–291, Sep. 2015, doi: 10.1016/j.isatra.2015.03.013.

[26] A. Belkhiri, A. Amouri, and A. Cherfia, "Design of fractional-order PID controller for trajectory tracking control of continuum robots," *FME Transactions*, vol. 51, no. 2, pp. 243–252, 2023, doi: 10.5937/fme2302243B.

[27] G. A. R. Ibraheem, A. T. Azar, I. K. Ibraheem, and A. J. Humaidi, "A novel design of a neural network-based fractional PID controller for mobile robots using hybridized fruit fly and particle swarm optimization," *Complexity*, vol. 2020, pp. 1–18, Apr. 2020, doi: 10.1155/2020/3067024.

[28] S. Gupta, A. P. Singh, D. Deb, and S. Ozana, “Kalman filter and variants for estimation in 2dof serial flexible link and joint using fractional order pid controller,” *Applied Sciences (Switzerland)*, vol. 11, no. 15, p. 6693, Jul. 2021, doi: 10.3390/app11156693.

[29] A. M. Abed *et al.*, “Trajectory tracking of differential drive mobile robots using fractional-order proportional-integral-derivative controller design tuned by an enhanced fruit fly optimization,” *Measurement and Control (United Kingdom)*, vol. 55, no. 3–4, pp. 209–226, Mar. 2022, doi: 10.1177/00202940221092134.

[30] P. Gallina, A. Rossi, and R. L. Williams, “Planar cable-direct-driven robots, part II: dynamics and control,” in *Proceedings of the ASME Design Engineering Technical Conference*, Sep. 2001, vol. 2, pp. 1241–1247, doi: 10.1115/detc2001/dac-21146.

[31] R. Eberhart and J. Kennedy, “Particle swarm optimization,” in *In: Proceedings of the IEEE international joint conference on neural networks*, 1995, vol. 4, no. 6, pp. 1942–1948.

[32] M. M. Noel, “A new gradient based particle swarm optimization algorithm for accurate computation of global minimum,” *Applied Soft Computing Journal*, vol. 12, no. 1, pp. 353–359, Jan. 2012, doi: 10.1016/j.asoc.2011.08.037.

[33] A. Merrad, A. Amouri, A. Cherfia, and S. Djeffal, “A reliable algorithm for obtaining all-inclusive inverse kinematics’ solutions and redundancy resolution of continuum robots,” *Arabian Journal for Science and Engineering*, vol. 48, no. 3, pp. 3351–3366, Mar. 2023, doi: 10.1007/s13369-022-07065-0.

## BIOGRAPHIES OF AUTHORS

**Hemama Aboud** was born in Oum El Bouaghi, Algeria, and graduated with a degree in Mechanical Engineering from the Faculty of Science and Technology, University of Larbi Ben M’hidi, Algeria, in 2000. She earned her master's degree in Robotics and Intelligent Systems from Larbi Ben M’hidi University, Oum El Bouaghi, Algeria, in 2009. In 2014, she was appointed as an assistant professor in the Department of Transport Engineering at the Faculty of Science and Technology, Frères Mentouri Constantine 1 University. Her research interests include robotics. She can be contacted at email: aboudha088@gmail.com.

**Ammar Amouri** received his Engineer's degree in Mechanical Engineering in 1995 and his Magister's degree in Mechanical Engineering and Robotics in 2011, both from Labri Ben M'Hidi University, Oum El Bouaghi, Algeria. He earned his Ph.D. in Mechanical Engineering from Frères Mentouri University Constantine 1, Algeria, in 2017. From 2013 to 2016, he served as a researcher at URMA/CRTI Research Center of Industrial Technologies in Annaba, Algeria. Currently, he is an Associate Professor, HDR, in the Department of Mechanical Engineering at Frères Mentouri University Constantine 1, Algeria. His primary research focus is on the design, modeling, and control of robotic systems. He can be contacted at email: ammar_amouri@yahoo.fr.

**Prof. Abdelhakim Cherfia** was born in 1963 in Algeria. He received the Ph.D. degree in Mechanical Engineering at the Frères Mentouri University Constantine 1, Algeria, in 2007. He is the head of the robotics and mechanical systems team at the Mechanics Laboratory of the Frères Mentouri University Constantine 1. Pr. Cherfia his main research interests is focused on the design and modeling of robotic systems. He can be contacted at email: cherfia_abdelhakim@yahoo.fr.

**Prof. Abdelaziz Mahmoud Bouchelaghem** was born in 1955 in Algeria. He received the Ph.D. in mechanical engineering at Technological University MGTU–STANKIN/Moscou (Russie) in 1986. His main research interest is focused on the mechanical manufacturing and design. He can be contacted at email: abdelaziz-mahmoud.bouchelaghem@univ-annaba.dz.