Robotica (2022), **40**, pp. 3863–3877
doi:10.1017/S0263574722000649

RESEARCH ARTICLE

Experimental study on the control of a suspended cable-driven parallel robot for object tracking purpose

Soroush Zare[1], Mohammad Reza Hairi Yazdi[1,3,*], Mehdi Tale Masouleh[2], Dan Zhang[3], Sahand Ajami[4] and Amirhossein Afkhami Ardekani[1]

[1]School of Mechanical Engineering, University of Tehran, Tehran, Iran, [2]Human and Robot Interaction Laboratory, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran, [3]Department of Mechanical Engineering, Lassonde School of Engineering, York University, Toronto, Canada, and [4]Mechanical Engineering Department, Amirkabir University of Technology, Tehran, Iran
*Corresponding author. E-mail: myazdi@ut.ac.ir

Received: 1 January 2022; **Revised:** 6 February 2022; **Accepted:** 30 March 2022; **First published online:** 17 May 2022

Keywords: cable-driven parallel robot, dynamics, neural network controller, PID, sliding mode control

Abstract
In this paper, control of a suspended cable-driven parallel robot has been experimentally investigated based on the dynamic model of the robot for object tracking purpose. In order to improve the tracking ability of the robot, three control approaches, namely kinematic PID, dynamic PD, and a kinematic sliding mode control (SMC), have been implemented, both on the Simscape and on the robot constructed at the Human and Robot Interaction Laboratory. Neural network controller and dynamic SMC have been implemented on the Simscape model. Afterward, the effectiveness of each approach has been investigated by employing the root mean square error (RMSE) index. Simulation and experimental results reveal the ability of each controller for precise and smooth control. For precise and real-time object tracking, YOLOv5-s and YOLOv4-tiny model are trained. By comparing the obtained index values, the kinematic PID demonstrates the best performance with the maximum RMSE value of 0.018 compared to other methods.

1. Introduction
Cable-driven parallel robot (CDPR) is a parallel robot in which rigid links are substituted by cables in order to control the end-effector (EE) position and orientation, called pose. This substitution leads to advantages such as reconfigurability [1], high payload to weight ratio, and large workspace besides ease of assembly, which are very useful for capturing videos or images in large zones with high precision [2]. A CDPR is completely constrained when the pose of its EE can be entirely determined if actuators are locked, and thus, all cable lengths are known. On the contrary, a CDPR is under-constrained when the number of actuators is less than the degrees of freedom. In an under-constrained CDPR, EE's pose is usually determined in the state in which the gravitational potential energy of the EE is minimized [3]; however, other solutions are possible. The inverse kinematic problem (IKP) consists of determining a vector of a joint variable for a prescribed pose of the EE. Determining the pose of EE from cable lengths is carried out by using the forward kinematics problem (FKP), in which its computation can be regarded as a troublesome drawback of under-constrained CDPRs. Several methods for calculating the FKP of parallel robots have been suggested, including numerical and analytical approaches. The problem can be simplified in several cases by identifying a set of algebraic equations which can be solved symbolically [4, 5]. Some algebraic formulations are obtained in some specific cases, resulting in polynomials of high degree which are extremely difficult to solve while also raising the computational cost. The direct geometric-static problem (DGP) is solved numerically for any kind of under-constrained CDPR, yielding all EE equilibrium configurations with a given geometry [6].

Solving the latter problem is a time-consuming process and needs a high-performance computer for such a specific configuration of CDPRs. In ref. [7], a specific method for under-constrained CDPRs based on combining neural networks (NNs) with the DGP is introduced which facilitates the real-time control of the robot. There are several different categories of NNs, each with its own set of use cases and degrees of complexity. The feed-forward structure is the simplest kind of NN, in which data flow from input to output along one unique direction. The back propagation NN, which is trained with supervision and uses the gradient-descent approach to minimize the squared error between the network output and the target one, is the most significant NN used in the literature [8]. Multilayer perceptron (MLP), the most popular network architecture, can be trained using specific input and output data, but even those which have not been trained can respond appropriately to a wide range of outputs. The first step in training a backpropagation MLP NN consists in choosing an appropriate NN architecture, which includes the hidden layers and the number of nodes for each layer. Cross-validation methods focused on the evaluation of error indices on various validation sets are used in particular to optimize the number of neurons while still preventing overfitting [9].

PD controllers usually are implemented to control CDPR [10]. In order to improve the performance of these types of controllers, feed-forward terms are used to predict the dynamic behavior of EE [11, 12]. Recently, sliding mode control (SMC) has been considered for control of CDPR for different purposes [13, 14]. This type of controller has a good performance against perturbations, but due to the use of the sign function, there are some discontinuities in the control input which cause the chattering phenomenon [15]. To reduce chattering, higher-order and gain adaptive SMC methods have been developed and implemented on a CDPR [16]. Finally, a sliding mode/linear (SML) controller is developed that not only reduces chattering but also compensates energy [17]. In ref. [18], the SML, a simple PD controller, and PD controller with a feed-forward term compensating the EE mass were experimentally compared. In ref. [19], a vision-based position control of planar CDPRs was experimentally implemented, and an online image processing procedure is developed for tracking the EE position. In ref. [20], an adaptive NN-based controller for a passive planar robot was investigated to drive serial robot links to track the desired reference model by the EE. In recent years, CDPRs have been used for rehabilitation purposes [21, 22]. In ref. [23], for rehabilitation purposes, especially arm disability, a feedback linearization approach is developed for maintaining tension in all the cables of the understudy CDPR. In ref. [24], a CDPR is used for surgical applications where recurrent NNs are used to increase the precision of tracking the path.

The main contribution of this paper consists in simulating the under-constrained CDPR in MATLAB® simulation environment [25], called Simscape. The Simscape model has been verified with the constructed robot for control purposes. Thereafter, the controllers consisting of kinematic PID, dynamic PID, and a kinematic SMC have been implemented, both on the Simscape and on the robot constructed at the Human and Robot Interaction Laboratory for object tracking purpose. Also, the NN controller and dynamic SMC have been implemented on the Simscape model.

The rest of the paper is organized as follows. In the next section, the configuration of the CDPR and formulation of each controller are summarized. Afterward, two models with two different versions were used for object tracking purpose. The mentioned controllers are implemented and compared through simulations and experiments. Finally, in the last section, the paper concludes with some hints and remarks as ongoing works.

2. The suspended CDPR

This section deals with the configuration of the 4-cable under-constrained CDPR and the relation to solve the IKP. Afterward, the dynamic model and control system of the robot are addressed.

2.1. Configuration of the 4-cable CDPR

Figure 1 illustrates the schematic of the CDPR constructed at the Human and Robot Interaction Laboratory, University of Tehran. Figure 2 depicts the schematic of the experimental setup. The

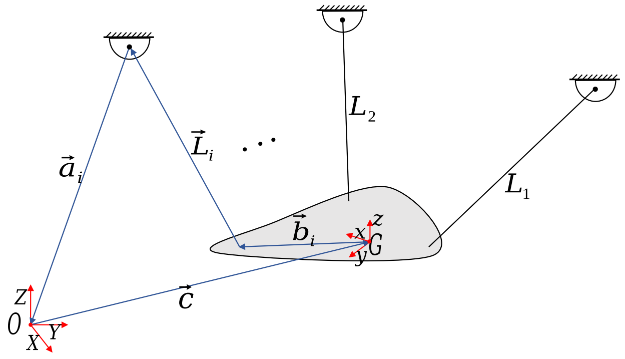


Figure 1. *The schematic of a cable-driven parallel robot with i cables.*

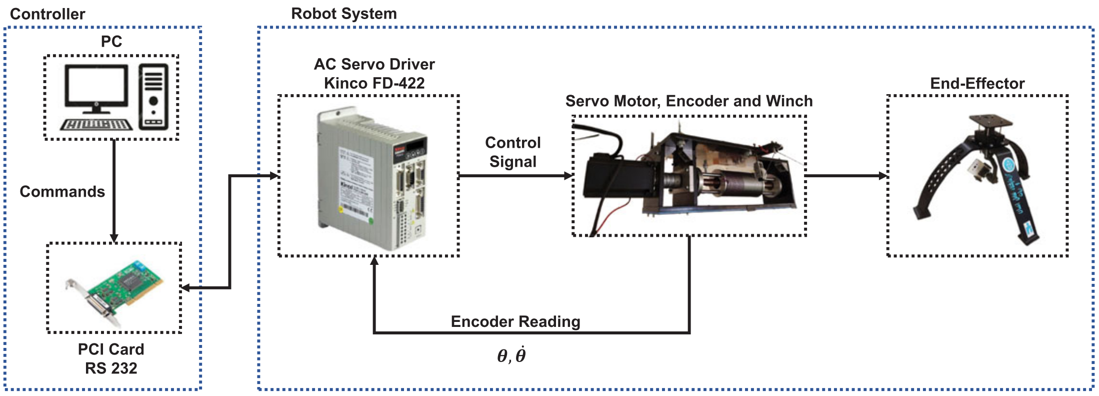


Figure 2. *The schematic of the experimental setup.*

functional workspace of the robot is a rectangular cube with dimensions of 405 cm × 285 cm × 300 cm [26]. It should be mentioned that the understudy robot is an improved and extended version of the robot presented in ref. [27]. The EE structure is built with 3D-printed legs and a special mechanism that not only keeps the stabilizer and camera in place but also protects them from unexpected jeopardy. Each actuator is coupled to a rolling mechanism and a 7.578 cm diameter winch, which is used to wind up and wind out each of the four cables. The initial length of each cable is 350 cm when it is placed in its home position. The kinematics equation of the robot can be written by the sum of the vectors as below [28]:

$$\boldsymbol{l_i} = \boldsymbol{c} + \boldsymbol{R}\boldsymbol{b_i} - \boldsymbol{a_i} \tag{1}$$

where R is the rotation matrix of the EE:

$$\boldsymbol{R} = \begin{bmatrix} c_\psi c_\phi & c_\psi s_\phi s_\theta - s_\psi c_\theta & c_\psi s_\phi c_\theta + s_\psi s_\theta \\ s_\psi c_\phi & s_\psi s_\phi s_\theta + c_\psi c_\theta & s_\psi s_\phi c_\theta - c_\psi s_\theta \\ -s_\phi & c_\phi s_\theta & c_\phi c_\theta \end{bmatrix} \tag{2}$$

where c and s stand for the cosine and sine of the indexed argument, respectively. Furthermore, θ, ϕ, and ψ signify as angles of rotation about x, y, and z-axis, respectively. In this case, $\|\mathbf{b}\|$ is 5 cm. Since $\|\mathbf{b}\| \ll \|\mathbf{l_i}\|$, it was assumed that the cables were connected to the middle of the plate of the EE. Therefore, Eq. (1) would be simplified as:

$$\mathbf{l_i} + \mathbf{a_i} = \mathbf{c} \tag{3}$$

whereby calculating $\mathbf{l_i}$ from the rotation of motors, the position of the EE can be determined:

$$\|\mathbf{l_i}\| = l_{0.i} - r\theta_i \tag{4}$$

where $l_{0.i}$ is the initial length of the i^{th} cable, r is the radius of the winch, and θ_i is the rotation of the i^{th} motor, read from the encoder of each AC servo motor. The AC servo motor model is Kinco SME80S-0075. From ref. [29], the dynamic equation of mobile platform can be expressed as follows:

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{g} = \boldsymbol{Jt} \tag{5}$$

where $\boldsymbol{x}$ is the vector of the pose of EE, and $\mathbf{J}$, $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{g}$ are the Jacobian matrix, inertia matrix, Coriolis matrix, and gravity vector, respectively. Also, $\mathbf{t}$ denotes cable tension. The dynamic equation of winches can be written as follows:

$$\mathbf{I_{mt}}\ddot{\boldsymbol{\theta}} + \boldsymbol{F_{vt}}\dot{\boldsymbol{\theta}} + \boldsymbol{F_{ct}}\,\text{sign}\,(\dot{\boldsymbol{\theta}}) + \mathbf{Nt} = \mathbf{u} \tag{6}$$

where $\mathbf{I_{mt}}$, $\boldsymbol{F_{vt}}$, and $\boldsymbol{F_{ct}}$ denote inertia matrix, viscous-friction coefficient, Coulomb friction, respectively. $\mathbf{u}$ represents the actuator torque. $\mathbf{N}$ denotes transmission ratio and can be defined as the relationship between cable length velocity $\dot{\boldsymbol{q}}$ and actuator angle velocity $\dot{\boldsymbol{\theta}}$ as follows:

$$\dot{\boldsymbol{q}} = \boldsymbol{N}\dot{\boldsymbol{\theta}} \tag{7}$$

where the relation between the initial length, $\boldsymbol{q^0} = \left[q_1{}^0 \;\; q_2{}^0 \;\; q_3{}^0 \;\; q_4{}^0\right]^T$, and the actuator angle, $\boldsymbol{\theta} = [\theta_1 \; \theta_2 \; \theta_3 \; \theta_4]^T$, can be expressed as:

$$\boldsymbol{q} = \boldsymbol{q^0} + \boldsymbol{N\theta} \tag{8}$$

The goal of controller consists in minimizing the error between the actuator angle $\boldsymbol{\theta}$ and the desired one, which is calculated from the IKP.

2.2. Control of the understudy CDPR

In this paper, three controllers, namely PID, sliding mode, and NNs controllers, are proposed which are applied to both kinematics and dynamics of the robot. In the kinematic control, the desired position of the EE is applied as the reference trajectory and the desired angles of rotation of the motors are calculated by the IKP. The velocity applied to the motor is the control signal, in which this type of controller is referred to as a kinematic-based controller [30]. In turn, when the torque applied to the motor is the control signal, a dynamic-based controller should be applied. Both types of controllers are implemented to minimize the error between the desired path and the actual path of the EE for a precise path tracking purpose.

2.2.1. PID controller

The PID controller can be formulated as:

$$u_m = K_p e_m + K_d \frac{de_m}{dt} + K_i \int_0^t e_m dt \quad m = 1, 2, 3, 4 \tag{9}$$

where m denotes the actuator's number, and e_m is defined as the difference between the desired and the actual position of actuator m. K_p, K_d, and K_i are the proportional, derivative, and integral gains, respectively. The goal of this controller consists in minimizing the tracking error, e, which is the difference between the actual and the desired path. These parameters have been tuned in the simulation environment using the standard method proposed in ref. [31], and due to the precision of the simulation, the same values have been used in the experiment. The obtained gains are given in Table I.

The dynamic PD controller is defined as in ref. [18], and the controller loop is shown in Fig. 3. The output signal of the controller is defined as follows:

$$\boldsymbol{\tau_c} = \boldsymbol{K_p e_\theta} + \boldsymbol{K_d e_{\dot{\theta}}} \tag{10}$$

Table I. *Kinematic PID controller gains.*

Gains	$\boldsymbol{K_p}$	$\boldsymbol{K_d}$	$\boldsymbol{K_i}$
Values	3	0.05	0.5

Table II. *Dynamic PD controller gains.*

Gains	$\boldsymbol{K_p}$	$\boldsymbol{K_d}$	$\boldsymbol{F_c}$	$\boldsymbol{F_v}$
Values	10	1	0.4	0.15

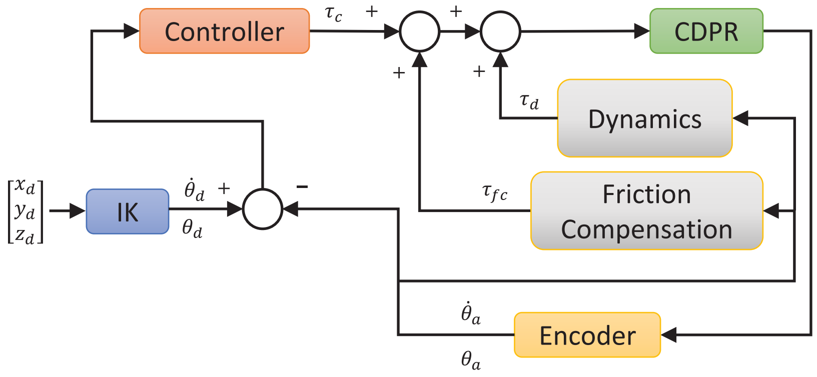


Figure 3. *The controller loop of dynamic PD.*

The $\boldsymbol{e_\theta}, \boldsymbol{e_{\dot{\theta}}}$ are the difference between the desired and actual actuator positions and velocities, respectively. Since each actuator is independently controlled, and the matrices $\boldsymbol{K_p}$ and $\boldsymbol{K_d}$ are diagonal, thus for ease of tuning, they have been tuned similarly for all actuators, which they can be written as follows:

$$K_p = K_{p,1} = K_{p,2} = K_{p,3} = K_{p,4} \tag{11}$$

$$K_d = K_{d,1} = K_{d,2} = K_{d,3} = K_{d,4} \tag{12}$$

The $\boldsymbol{\tau_d}$ is the feed-forward torque, and the $\boldsymbol{\tau_{fc}}$, the friction compensation, is considered to compensate the friction and defined as follows:

$$\boldsymbol{\tau_{fc}} = \boldsymbol{F_c}\,\mathrm{sign}\,(\boldsymbol{e_{\dot{\theta}}}) + \boldsymbol{F_v}\boldsymbol{e_{\dot{\theta}}} \tag{13}$$

The $\boldsymbol{F_c}$ and $\boldsymbol{F_v}$ are considered static and viscous coefficients of each motor, respectively. As each motor is independently controlled, these two matrices are diagonal and are evaluated particularly for the designed EE. These parameters are determined for each motor to compensate static friction at the beginning of the motion, which leads to a smooth motion in tracking the path. As there was no significant difference between the two parameters for four motors, these two parameters are chosen the same for all motors as follows:

$$F_c = F_{c,1} = F_{c,2} = F_{c,3} = F_{c,4} \tag{14}$$

$$F_v = F_{v,1} = F_{v,2} = F_{v,3} = F_{v,4} \tag{15}$$

The controller gains have been tuned to achieve accuracy and stability with the mentioned method in tuning the previous kinematic PID controller, and the gains are given in Table II.

Table III. *Kinematic SMC controller gains.*

Gains	K_1	K_2	η
Values	80	12	2.4

2.2.2. SMC

For the kinematics approach, the error and sliding surface and the controller are expressed as:

$$\begin{cases} e_i = \theta_i - \theta_i^* \\ \dot{e}_i = \dot{\theta}_i - \dot{\theta}_i^* \end{cases} \quad i = 1, 2, 3, 4 \tag{16}$$

$$S = k_1 e_i + k_2 \dot{e}_i = k_1 x_1 + k_2 x_2 \tag{17}$$

where θ, θ^* are the desired position and velocity of the actuators, respectively. According to the sliding surface, if the condition $k_1 k_2 > 0$ is satisfied, the system would be exponentially stable. The following is the state space of the system, which includes modeling uncertainties:

$$\begin{cases} x_1 = e_i \\ x_2 = \dot{e}_i \end{cases} \Rightarrow \begin{cases} \dot{x}_1 = x_2 + \alpha_1 \\ \dot{x}_2 = u + \alpha_2 \end{cases} \tag{18}$$

The above model takes into account model uncertainties and noises in α_1 and α_2. Furthermore, the desired time derivative of the sliding surface has been chosen as:

$$\dot{S} = -\eta^* \operatorname{sign}(S) \tag{19}$$

which by some substitution and simplification, the control effort signal is derived as follows:

$$u = -\frac{K_1}{K_2} x_2 - \frac{\eta^*}{K_2} \operatorname{sign}(S) \tag{20}$$

where the following inequality has been obtained using α_1 and α_2 as the bounds for $\tilde{\alpha}_1$ and $\tilde{a}_2$, respectively; η is a positive scalar:

$$|k_1 \alpha_1 + k_2 \alpha_2| \leq k_1 \tilde{\alpha}_1 + k_2 \tilde{\alpha}_2 = \alpha^* \tag{21}$$

$$\eta^* = \eta + \alpha^* \tag{22}$$

The Lyapunov candidate function is chosen as:

$$V = \frac{1}{2} S^2 \tag{23}$$

The following results were obtained by differentiating the proposed Lyapunov function with respect to time:

$$\dot{V} = S\dot{S} \leq -S \times \eta \operatorname{sign}(S) = -\eta |S| \tag{24}$$

As a result, the SMC's ability to asymptotically stabilize the system has been proven. To prevent chattering, the implemented control signal is as follows:

$$u = \frac{-K_1}{K_2} \dot{e} - \frac{\eta^*}{K_2} \tanh(S) \tag{25}$$

The gains have been tuned with the same method of the PID controllers, and the obtained gains are given in Table III.

In the dynamic control, the desired position of the EE is applied as the reference trajectory and the desired angles of rotation of the motors are calculated by the IKP. For the dynamics approach, the sliding surface and the controller can be expressed as [18]:

$$\dot{\tau}_{m.i} = -K_1 [\sigma_i]^{\frac{\alpha}{2-\alpha}} - K_2 [\dot{\sigma}_i]^{\alpha} \tag{26}$$

Table IV. *Dynamic SMC controller gains.*

Gains	K_1	K_2	β	ε_σ	$\dot{\varepsilon}_\sigma$
Values	−30	−15	1.05	40	100

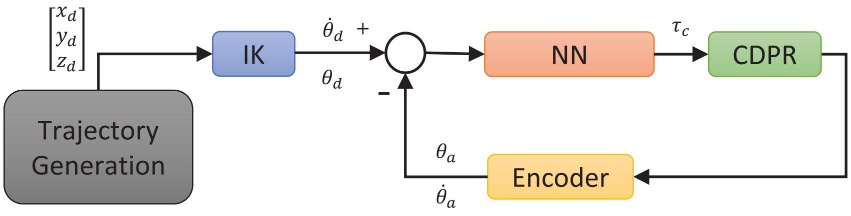


Figure 4. *The schematic of the neural network controller.*

where K_1, K_2 are the controller gains and $\dot{\tau}_{m.i}$ is $\dot{\tau}_m$ for the i^{th} motor and $[\sigma_i]^\alpha$ can be derived as:

$$\lceil \sigma_i \rfloor^\alpha = |\sigma_i|^\alpha \text{ sign } (\sigma_i) \tag{27}$$

$$\alpha = \max\left(1 - \beta\left(\frac{|\sigma_i|}{|\sigma_i| + \varepsilon_\sigma} + \frac{|\dot{\sigma}_i|}{|\dot{\sigma}_i| + \varepsilon_{\dot{\sigma}}}\right), 0\right) \tag{28}$$

where σ, β, ε_σ, and $\dot{\varepsilon}_\sigma$ are constant parameters presented in ref. [18]. The torque applied to the motor was the control signal which was the time integral of Eq. (26). The gains have been tuned with the same method of the previous controllers, and the obtained gains are given in Table IV.

2.2.3. NN controller

For the MLP NN, the optimal design with the minimal local root mean square error (RMSE) is discovered by grid search. The MLP NNs in which the architecture with hidden layers is $64 \times 32 \times 16 \times 8 \times 4$ have been chosen and trained by the collected data. The schematic of the NN controller is shown in Fig. 4. The error between the desired location and the actual one and its derivative are chosen as the input of the NN, and the torque which has to be exerted to each motor is the output. These data have been obtained by different trajectories, including helix, snail, circle, straight line, with the different equations, and also different points. SGD, Adam, Momentum, L-BFGS, and Levenberg–Marquardt are among the well-known optimization approaches and have been chosen. Levenberg–Marquardt, a second-order optimization which correctly considers the nonlinearity of equations, leads to the least RMSE. Also, the hyperbolic tangent activation function produced the least RMSE in comparison with the linear and the sigmoid activation functions.

3. Object tracking

For finding object coordinates in soccer pitch, a four-point homography parameterization which maps the four corners from one image into the second image was used. After detecting four corners of the pitch, a homography matrix was computed, and then, the pitch image was transformed by the foregoing matrix in order to display the image in the form of the birds-eye view. This transformation makes distortion in the image and also makes some difficulties for detecting objects in the image with traditional methods. Moreover, using color for object detection was not as precise as it should be, and also, the EE of the robot could be confused in the case of observing two objects with the same color. Another challenging issue is that the ball can move fast and cause motion blur or even partially occluded by players [32]. For these reasons, the YOLO algorithm was used for object detection in the transformed image. After the region-based convolutional neural network [33], the YOLO [34] algorithm was developed to speed up the object detection process. YOLO algorithm, which has improved in many aspects such as accuracy

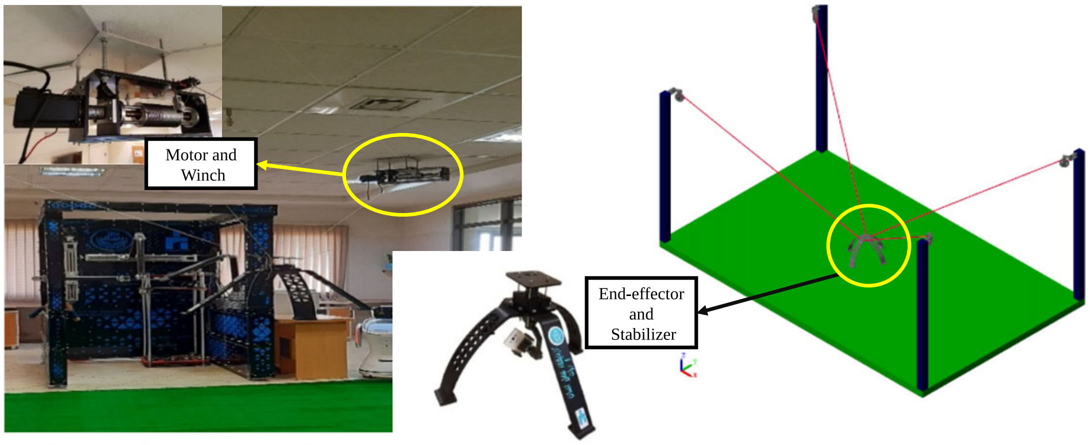


Figure 5. *The constructed and Simscape model of the CDPR.*

and speed, is an extremely fast, one-stage, and end-to-end algorithm for object detection. Two models with two different versions were used for detecting the object, which EE had to follow. The YOLOv5-s model was run in PyTorch and YOLOv4-tiny [35], which was compiled on Darknet. Both models are fast and do not use much amount of memory which makes them suitable for real-time object detection tasks which need pace in detection. Since only one object was desired to be tracked, two light models were utilized for object detection on the provided dataset, in which a comparison is performed in terms of speed and accuracy between the two models. The dataset used for training models has contained 823 training images and 175 testing images in which all the images were transformed to birds-eye view with a resolution of 486 × 342. This dataset was collected with different variations containing different balls concerning sizes, colors, static, and blurred due to a motion. All of the images were manually annotated with bounding boxes for YOLO for only detecting one specific object (static and blurred due to a motion). In order to avoid training the model from scratch, transfer learning [36] was applied for training both models, and they were fine-tuned using pre-trained checkpoints on the COCO dataset [37]. The hyperparameters of the models are as follows: For training the YOLOv4-tiny model, the batch size, momentum, and weight decay were set to 16, 0.9, and 0.0005, respectively. Training of YOLOv4-tiny was performed in 2000 iterations from which the one with the smallest loss was selected, and the initial learning rate was set equal to 0.002. Also, the YOLOv5-s model was trained in 50 epochs with batch size 64, momentum 0.937, weight decay 0.0005, and an initial learning rate 0.01. Precision, recall, mean average precision, F1 score, and inference time were selected as indicators. The processing speed should exceed 30 frames per second for real-time object detection purposes.

4. Results

4.1. Simulation

Figure 5 depicts the 4-cable CDPR constructed at Human and Robot Interaction Laboratory, University of Tehran and Simscape model, which is built in MATLAB® Simulink to enforce and validate the derived analytical formulas and designed controllers in this section. This section investigates the robot's ability to track a nonlinear path using the suggested controllers' effort signal. For proper monitoring, the suggested trajectory is a three-dimensional direction that necessitates different torque values from each actuator. The path's parametric formulation, which was used both in simulation and in practice, is shown in Fig. 6 and is as follows:

$$\begin{aligned} x &= 0.05t \sin t \\ y &= 0.05t \cos t \\ z &= 0.05t \end{aligned} \tag{29}$$

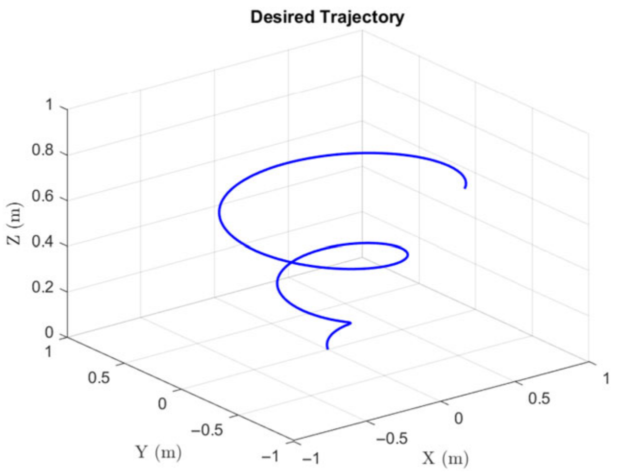


Figure 6. *The three-dimensional presentation of the path under study.*

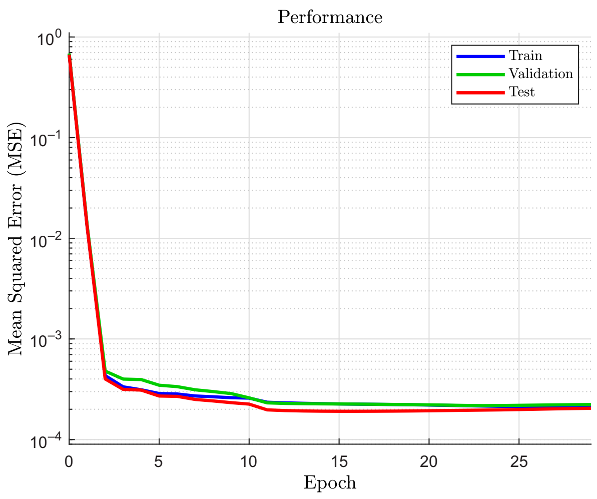


Figure 7. *The performance of the trained neural network.*

In order to examine the ability of the controller and make the simulation more realistic to the built CDPR, stiffness, friction of the winch, and the friction of the pulley have been determined and modeled. The Simscape model also determined the angle of the cable with respect to the winch for each motor. Furthermore, angles of rotation of EE about the x, y, and z-axis also have been simulated and can be monitored during the simulation. Figure 7 demonstrates the performance of the trained NN for the NN controller. If the model receives insufficient training, it will underfit the train and test sets. If the model receives too much training, it will overfit the training dataset and perform poorly on the test set. A reasonable compromise is to train on the training dataset but stop when performance on the validation dataset begins to deteriorate. In order to prevent overfitting and improve NN ability to predict on non-training data, validation stopping is implemented, and the training stops after four consecutive increases in validation error, and the best performance is taken from the epoch with the lowest validation error [38]. The results of the simulation and the RMSE index for each controller are shown in Fig. 8 and Table V, respectively. According to the results provided from the RMSE index, all of the controllers track the path properly with a maximum RMSE of 0.072, which occurred in kinematics SMC. Due to the inherent robustness of the SMC algorithm, it tracked the path more smoothly but less precisely. Also,

Table V. *The RMSE index for controlling method in simulation.*

Controller	X	Y	Z
Kinematic PID	0.035	0.058	0.010
Dynamic PD	0.007	0.037	0.012
Kinematic SMC	0.049	0.072	0.012
Dynamic SMC	0.006	0.038	0.011
Dynamic NN	0.007	0.037	0.012

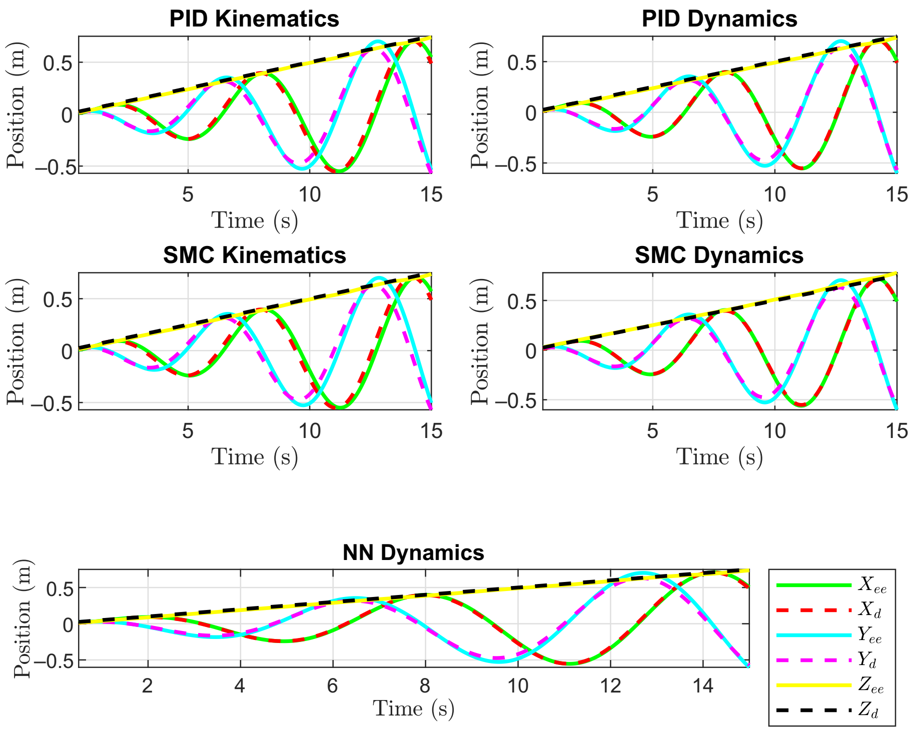


Figure 8. *Simulation results of the five controllers for tracking the path along the x, y, and z axes.*

the RMSE index demonstrates that all dynamic controllers similarly tracked the path and had a better performance than the kinematic controllers.

4.2. Experimental

Figure 9 depicts the results of each controller implemented on the built CDPR. From the result provided by the RMSE index in Table VI, it can be concluded that all of the controllers tracked the path properly with the maximum RMSE of 0.029, which occurred in dynamic PID. The kinematic PID, which was selected for object tracking, had a better performance than other controllers. It should be mentioned that in the Simscape model, a revolute joint was used for simulating the angle of cable with respect to the winch, which decreases the accuracy of the controller. Due to mentioned reason, the accuracy of the simulation is slightly less than practical.

Figure 10 represents the training results for both YOLO models for detecting the big red ball. According to the obtained results, both models detected the defined object precisely. The small value of

Table VI. *The RMSE index for controlling method in practical.*

Controller	X	Y	Z
Kinematic PID	0.018	0.017	0.008
Dynamic PD	0.020	0.029	0.016
Kinematic SMC	0.021	0.022	0.005

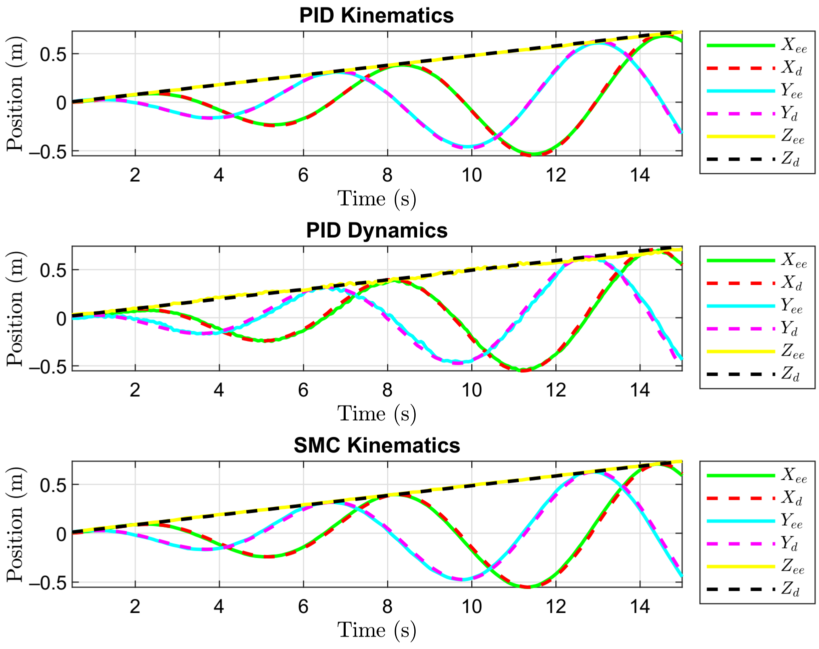


Figure 9. *Practical results of the three controllers for tracking the path along the x, y, and z axes.*

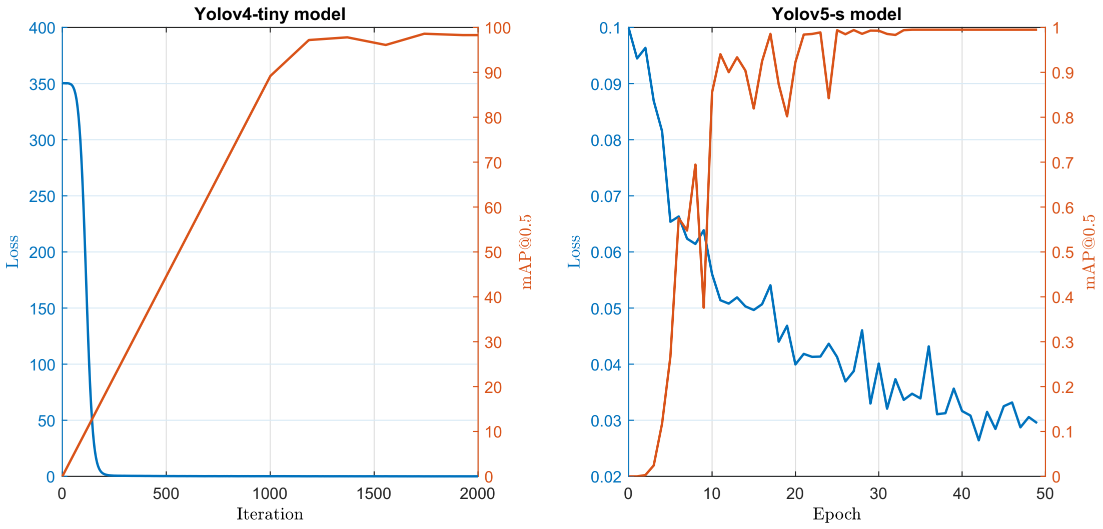
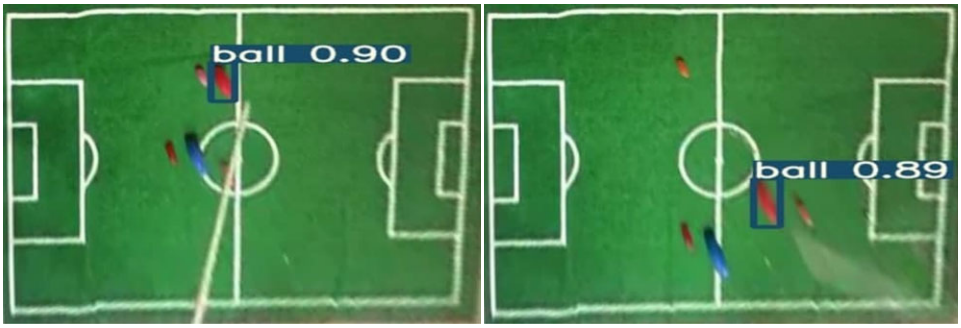


Figure 10. *The training results of the YOLO models.*

Table VII. *The RMSE index of each encoders for object tracking.*

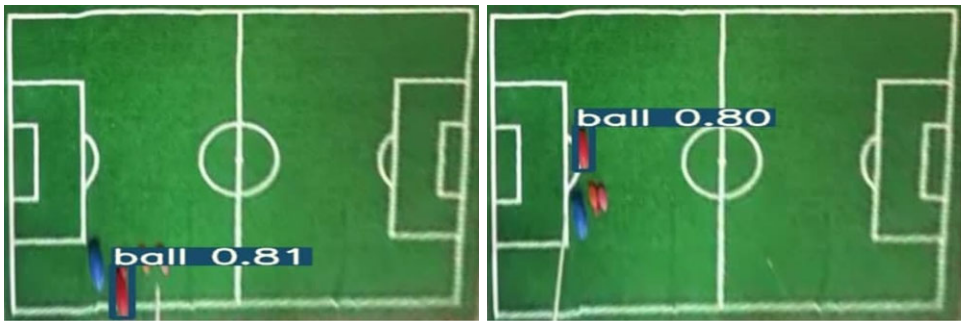
Index	$\theta 1$	θ_2	θ_3	θ_4
RMSE	0.0166	0.0014	0.0708	0.0049

(a)



YOLOv4-tiny.

(b)



YOLOv5-s.

Figure 11. *YOLO detection results.*

loss shows that the model is well trained. On the other hand, the high mAP value indicates that the model has a good performance. However, YOLOv4-tiny has a higher confidence in detecting the object and is more robust. On the other hand, YOLOv5-s has higher FPS on CPU, but not on GPU; therefore, for running on CPU, YOLOv5-s is a better choice according to its detection speed, but for running on a GPU, YOLOv4-tiny overtakes YOLOv5-s. From the mAP and F1 score, it can be inferred that YOLOv5-s has slightly better performance than YOLOv4-tiny. The best mAP values of YOLOv5-s and YOLOv4-tiny are 99.5% and 98.5%, respectively.

From this result, it can be concluded that both models approximately have the same training capability on the provided small dataset for detecting one object. The sample of detection results of both YOLO models is shown in Fig. 11. As aforementioned, the kinematic PID has been chosen as the controller for object tracking. The rotation angle of the four motors, taken from the encoders, is shown in Fig. 12. The RMSE index errors are given in Table VII for the motors of one to four, respectively, which confirm the ability of the controller for tracking the path precisely. It should be mentioned that during the object tracking, while the camera has been covered, the location of EE is the last location for determining object location. Once the camera determines the object's new location, the EE tries to track it as fast as it can. Some jumps in Fig. 12 are occurred due to the latter reason.

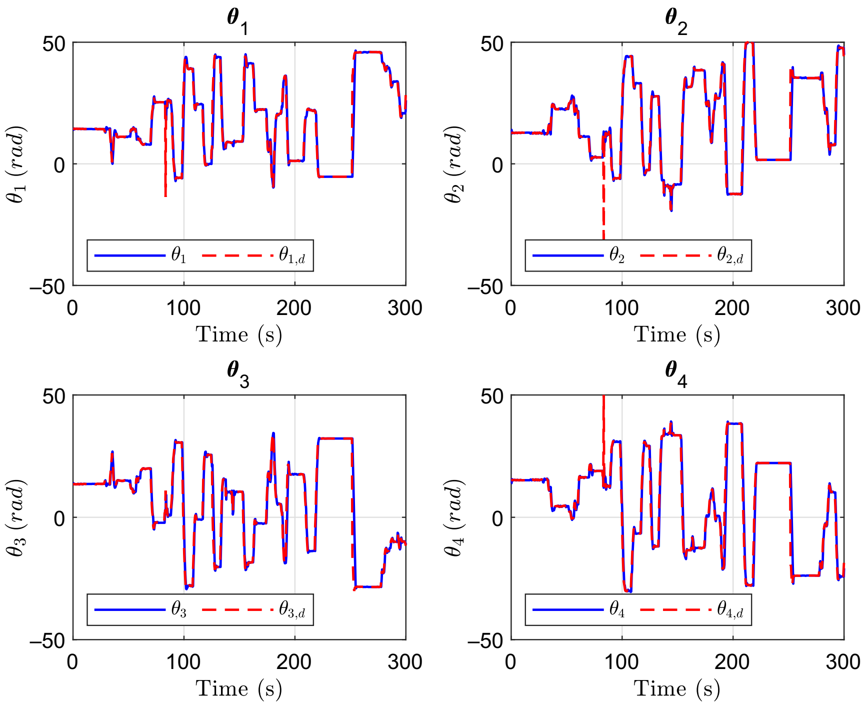


Figure 12. *The output of each encoder for object tracking.*

5. Conclusion

This paper investigated the evaluation of different controllers in both simulation and practice in which two methods for tracking objects were proposed. In the simulation, the model of the robot was verified, which made possible the validation of the proposed controllers or other studies in simulation. Moreover, a novel NN controller based on the dynamic model of an under-constrained CDPR was proposed. The results showed that the proposed controllers exhibited satisfactory performance on the predefined trajectory in simulation and experiment. Furthermore, for tracking purpose we resorted to the so-called YOLOv5-s as it was faster than YOLOv4-tiny in object detection on CPU. Future works aim at using deep reinforcement learning strategies for controlling an under-constrained CDPR.

Acknowledgement. None.

Conflicts of interest. None.

Financial support. None.

Ethical considerations. None.

Authors' contributions. Soroush Zare conceived and designed the study under co-supervision of Mohammad Reza Hairi Yazdi and Mehdi Tale Masouleh. All others are coauthors.

References

[1] L. Gagliardini, S. Caro, M. Gouttefarde and A. Girin, "Discrete reconfiguration planning for cable-driven parallel robots," *Mech. Mach. Theory* **100**, 313–337 (2016).

[2] S. Zare, M. Ghanatian, M. R. H. Yazdi and M. T. Masouleh, "Reconstructing 3-Constrained Graphical Model Using an Under-Driven Cable-Constrained Parallel Robot," **In:** *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)* (IEEE, 2020) pp. 1–6.

[3] M. Carricato and J.-P. Merlet, “Stability analysis of underconstrained cable-driven parallel robots,” *IEEE Trans. Robot.* **29**(1), 288–296 (2012).

[4] M. Carricato and J.-P. Merlet, “Geometrico-Driven Analysis of Under-Constrained Cable-Driven Parallel Robots,” **In:** *Advances in Robot Kinematics: Motion in Man and Machine* (Springer, 2010) pp. 309–319.

[5] P. Bosscher, R. L. Williams II, L. S. Bryson and D. Castro-Lacouture, “Cable-suspended robotic contour crafting system,” *Automat. Constr.* **17**(1), 45–55 (2007).

[6] G. Abbasnejad and M. Carricato, “Direct geometrico-static problem of underconstrained cable-driven parallel robots with *n* cables,” *IEEE Trans. Robot.* **31**(2), 468–478 (2015).

[7] S. Zare, M. S. Haghighi, M. R. H. Yazdi, A. Kalhor and M. T. Masouleh, “Kinematic Analysis of an Under-Suspended Cable-Dimension Robot Using Neural Networks,” **In:** *2020 28th Iranian Conference on Electrical Engineering (ICEE) (IEEE, 2020)* pp. 1–6.

[8] L. Medsker, “Design and Development of Hybrid Neural Network and Expert Systems,” **In:** *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN’94)* (IEEE, vol. **3**, 1994) pp. 1470–1474.

[9] A. Gholipour, B. N. Araabi and C. Lucas, “Predicting chaotic time series using neural and neurofuzzy models: A comparative study,” *Neural Process. Lett.* **24**(3), 217–239 (2006).

[10] S. Kawamura, H. Kino and C. Won, “High-speed manipulation by using parallel wire-driven robots,” *Robotica* **18**(1), 13–21 (2000).

[11] J. Lamaury and M. Gouttefarde, “Control of A Large Redundantly Actuated Cable-Driven Parallel Robot,” **In:** *2013 IEEE International Conference on Robotics and Automation* (IEEE, 2013) pp. 4659–4664.

[12] J. C. Santos, A. Chemori and M. Gouttefarde, “Model Predictive Control of Large-Free Cable-Driven Parallel Robots,” **In:** *International Conference on Cable-Driven Parallel Robots* (Springer, 2019) pp. 221–232.

[13] M. Zeinali and A. Khajepour, “Design and Application of Chattering-Actuated Sliding Mode Controller to Cable-Driven Parallel Robot Manipulator: Theory and Experiment,” **In:** *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (vol. **44106**, 2010) pp. 319–327.

[14] G. El-Ghazaly, M. Gouttefarde and V. Creuze, “Adaptive Terminal Sliding Mode Control of A Redundantly-Based Cable-Twisting Parallel Manipulator: Cogiro,” **In:** *Cable-Driven Parallel Robots* (Springer, 2015) pp. 179–200.

[15] V. I. Utkin, *Sliding Modes in Control and Optimization* (Springer Science & Business Media, Verlog Berlin, 2013).

[16] C. Schenk, C. Masone, A. Pott and H. H. Bülthoff, “Application of A Differentiator-Driven Adaptive Super-Driven Controller for A Redundant Cable-Dof Parallel Robot,” **In:** *Cable-Driven Parallel Robots* (Springer, 2018) pp. 254–267.

[17] E. Tahoumi, F. Plestan, M. Ghanes and J.-P. Barbot, “A Controller Switching Between Twisting and Linear Algorithms for an Electropneumatic Actuator,” **In:** *2018 European Control Conference (ECC)* (IEEE, 2018) pp. 2368–2373.

[18] E. Picard, E. Tahoumi, F. Plestan, S.éphane Caro and F. Claveau, “A New Control Scheme of Cable-Driven Parallel Robot Balancing Between Sliding Mode and Linear Feedback,” **In:** *The 21st IFAC World Congress (IFAC∼ 2020)* (2020).

[19] H. Bayani, M. T. Masouleh and A. Kalhor, “An experimental study on the vision-based control and identification of planar cable-driven parallel robots,” *Robot. Auton. Syst.* **75**(4), 187–202 (2016).

[20] V. Bahrami, A. Kalhor and M. T. Masouleh, “Dynamic model estimating and designing controller for the 2-DoF planar robot in interaction with cable-driven robot based on adaptive neural network,” *J. Intell. Fuzzy Syst.* **Preprint**(1), 1–20 (2021).

[21] A. M. Khomami and F. Najafi, “A survey on soft lower limb cable-driven wearable robots without rigid links and joints,” *Robot. Auton. Syst.* **144**(1), 103846 (2021).

[22] S. M. Youssef, M. A. Soliman, M. A. Saleh, M. A. Mousa, M. Elsamanty and A. G. Radwan, “Underwater soft robotics: A review of bioinspiration in design, actuation, modeling, and control,” *Micromachines* **13**(1), 110 (2022).

[23] V. Bahrami, A. Kalhor and M. T. Masouleh, “Dynamic modeling and design of controller for the 2-driven derial chain actuated by a cable-CNN robot based on feedback linearization,” *Proc. Inst. Mech. Eng. C J. Mech. Eng. Sci.*, 09544062211027922 (2021).

[24] M. Hwang, B. Thananjeyan, S. Paradis, D. Seita, J. Ichnowski, D. Fer, T. Low and K. Goldberg, “Efficiently calibrating cable-driven surgical robots with RGBD fiducial sensing and recurrent neural networks,” *IEEE Robot. Automat. Lett.* **5**(4), 5937–5944 (2020).

[25] MATLAB, *version 9.10.0 (R2021a)* (The MathWorks Inc., Natick, Massachusetts, 2021).

[26] R. Mersi, S. Vali, G. Abbasnejad, M. Tale and etal Masouleh, “Design and Control of A Suspended Cable-Time Parallel Robot with Four Cables,” **In:** *2018 6th RSI International Conference on Robotics and Mechatronics (IcRoM)* (IEEE, 2018) pp. 470–475.

[27] M. Zarei, A. Aflakian, A. Kalhor and M. T. Masouleh, “Oscillation damping of nonlinear control systems based on the phase trajectory length concept: An experimental case study on a cable-driven parallel robot,” *Mech. Mach. Theory* **126**(2), 377–396 (2018).

[28] A. Aflakian, A. Safaryazdi, M. T. Masouleh and A. Kalhor, “Experimental study on the kinematic control of a cable suspended parallel robot for object tracking purpose,” *Mechatronics* **50**(10), 160–176 (2018).

[29] W. Shang, F. Xie, B. Zhang, S. Cong and Z. Li, “Adaptive cross-coupled control of cable-driven parallel robots with model uncertainties,” *IEEE Robot. Automat. Lett.* **5**(3), 4110–4117 (2020).

[30] S. Qian, B. Zi, W.-W. Shang and Q.-S. Xu, “A review on cable-driven parallel robots,” *Chin. J. Mech. Eng.* **31**(1), 1–11 (2018).

[31] J. Ziegler and N. Nichols, “Optimum settings for automatic controllers,” *InTech-Int. J. Meas. Cont.* **42**(6), 94–101 (1995).

[32] M. Pobar and M. Ivašić-Kos, “Detection of the Leading Player in Handball Scenes Using Mask R-CNN and Stips,” **In:** *Eleventh International Conference on Machine Vision (ICMV 2018)* (International Society for Optics and Photonics, vol. **11041**, 2019) pp. 110411V.

[33] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *IEEE Trans. Patt. Anal.* **38**(1), 142–158 (2015).
[34] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-time Object Detection,” **In:** *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 779–788.
[35] Z. Jiang, L. Zhao, S. Li and Y. Jia, Real-time object detection method based on improved yolov4-tiny. arXiv preprint arXiv: 2011. 04244 (2020).
[36] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, “Feature Pyramid Networks for Object Detection,” **In:** *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017) pp. 2117–2125.
[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” **In:** *European Conference on Computer Vision* (Springer, 2014) pp. 740–755.
[38] X. Ying, “An Overview of Overfitting and Its Solutions,” **In:** *Journal of Physics: Conference Series* (IOP Publishing, vol. **1168**, 2019) pp. 022022.

Cite this article: S. Zare, M. R. H. Yazdi, M. T. Masouleh, D. Zhang, S. Ajami and A. Afkhami Ardekani (2022). “Experimental study on the control of a suspended cable-driven parallel robot for object tracking purpose”, Robotica **40**, 3863–3877. https://doi.org/10.1017/S0263574722000649