


## Article

# A Non-Linear Continuous-Time Generalized Predictive Control for a Planar Cable-Driven Parallel Robot

Fouad Inel <sup>1,\*</sup>, Ali Medjbouri <sup>1</sup> and Giuseppe Carbone <sup>2,\*</sup> <sup>1</sup> Department of Mechanical Engineering, Automatic Laboratory, University of Skikda, 21000 Skikda, Algeria; inelfouad@yahoo.fr<sup>2</sup> DIMEG, University of Calabria, Via Bucci Cubo 45C, 87036 Rende, Italy

\* Correspondence: f.innel@univ-skikda.dz (F.I.); giuseppe.carbone@unical.it (G.C.)

**Abstract:** This paper addresses a novel nonlinear algorithm for the trajectory tracking of a planar cable-driven parallel robot. In particular, we outline a nonlinear continuous-time generalized predictive control (NCGPC). The proposed controller design is based on the finite horizon continuous-time minimization of a quadratic predicted cost function. The tracking error in the receding horizon is approximated using a Taylor-series expansion. The main advantage of the proposed NCGPC is based on using an analytic solution, which can be truncated to a desired degree of order of the Taylor-series. This allows us to achieve a prediction horizon of NCGPC tracking performance. The description of the proposed NCGPC method is followed by a comparison between NCGPC and a conventional computed torque control (CTC) method. Robustness tests are performed by considering payload and parameter uncertainties for both controllers. Simulation results of NCGPC compared to the commonly used CTC prove the effectiveness and advantages of the proposed approach.



**Citation:** Inel, F.; Medjbouri, A.; Carbone, G. A Non-Linear Continuous-Time Generalized Predictive Control for a Planar Cable-Driven Parallel Robot. *Actuators* **2021**, *10*, 97. <https://doi.org/10.3390/act10050097>

Academic Editors: Ioan Doroftei, Gianluca Rizzello and Karsten Berns

Received: 13 February 2021

Accepted: 30 April 2021

Published: 4 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** planar cable-driven parallel robot; nonlinear models; continuous-time generalized predictive control; control robustness; tracking performance

## 1. Introduction

Cable-driven parallel robots (CDPRs) are a class of parallel robots where the end-effector is supported by a system of cables, which are extended or retracted by using motors and pulleys. Recently, CDPRs are widely used for several applications, such as lifting heavy loads and rehabilitation applications, as reported, for example, by Albus et al. [1], Tang [2], Pinto et al. [3], and Morizono et al. [4]. The challenges arising in the control of CDPRs have been the subject of many research works. Particularly significant in this field are, for example, the work on orientability by Vikranth Reddy [5]; the work [6] where the control takes into account the viscoelastic effect of cables; the work on the pseudo-drag effect at high speed that is reported, for example, in [7]; or the work on high-frequency and oscillation reduction that is reported, for example, in [8].

The accuracy of a cable-driven parallel robot can be strongly affected by the implemented control strategy. In this paper, we propose a robust control strategy for the real-time control of a cable-driven robot. In recent decades, predictive control has become one of the most important research topics in nonlinear control systems, such as that reported by Koreyam et al. for ICaSbot CDPR in [9], or by considerations on the feasibility for enhanced control strategies, as reported in [10]. Model predictive control (MPC), also known as receding horizon control, is generally based on a priori knowledge of the process via a prediction model ensuring the estimation of the future outputs. This can be achieved by means of frequency analyses, as proposed, for example, in [11], or by establishing proper dynamic models with cable elasticity, as reported, for example, in [12]. Once this model has been determined, the MPC algorithm can be formulated as an online optimization problem on a finite moving horizon. At each sampling step, the controller tries to minimize the cost functions subject to specific constraints. Constraints can be set on the feedback

linearization, as proposed, for example, in [13]; they can be based on adaptative control strategies, as proposed, for example, in [14]; or they can be defined on elastic models of the actuators, as proposed, for example, in [15]. When the control sequence is found, only the first component is applied to the process. At the next sampling time, the prediction horizon moves one step forward, and the optimization problem is repeated iteratively. However, the above-mentioned problem becomes more difficult, or even impossible to solve, when it is implemented for nonlinear systems with fast dynamics, especially in the case where several constraints are considered, such as that proposed by Inel et al. in [16] or by Carbone et al. in [17]. This is because the on-line nonlinear optimization algorithm imposes a heavy computational burden which requires a large computation time, as reported, for example, by Tang and Shao in [18], by Li et al. in [19], or by Poignet and Gautier in [20]. Therefore, in most cases, linear models are used to predict the future behavior of the nonlinear system over a finite horizon. The advantage of using linearized prediction models lies in the fact that in the absence of constraints, the optimal solution can be written explicitly. Nevertheless, the robustness of the control law depends directly on considering the nonlinear nature of the system. Nonlinear model predictive control (NMPC) is, therefore, a powerful strategy. The best way to address these difficulties arising in the design and implantation of NMPC is to adopt an analytic approach, where the control problem is formulated in a closed form to avoid online optimization with the approach proposed, for example, in [21] or the application proposed by Hedjar et al. in [22].

Non-linear continuous-time generalized predictive control (NCGPC) was first proposed in [23] for linear plants. It has similar properties to those of the discrete time generalized predictive control (GPC), and it is a promising alternative for dealing with fast dynamic nonlinear systems, as mentioned, for example, in [24]. In general, the predictive control problem has specific advantages for solving—in real time—a finite horizon closed-loop optimal control subject to the system dynamic constraints involving states and controls. The main advantage of MPC is the fact that it allows the current timeslot to be optimized while taking future timeslots into account. This is achieved by optimizing a finite time horizon while only implementing the current timeslot and then optimizing again for the next timeslot.

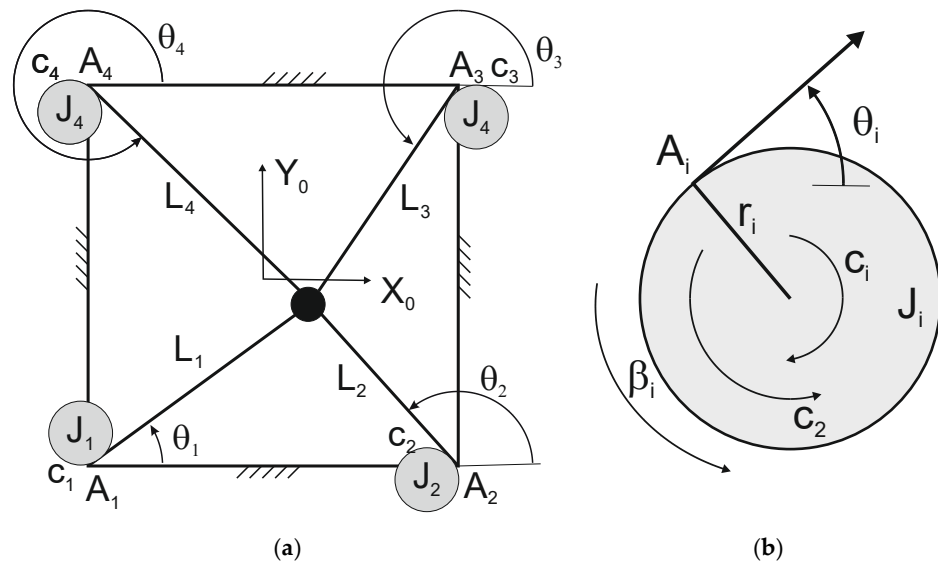
The NCGPC prediction model is based on Taylor series expansion. However, obtaining an analytic solution imposes a constraint on the order of the Taylor-series expansion to the corresponding output relative degree [25,26]. Most work in generalized predictive control is based on a discrete-time approach, but one can use a transfer function approach to achieve a continuous-time approach. Note that this control strategy requires the stability of all zero dynamics, and the closed-loop system stability is guaranteed for a relative degree less than or equal to four.

The aim of this work is the application and implementation of an NCGPC control algorithm for the motion control of a planar four-cable-driven parallel robot. The proposed control strategy is based on a generalized predictive control with a discrete-time approach, including dynamic constraints. The predictive error is obtained in advance by expanding the output and reference signals in an  $r_i$ -th order Taylor series, where  $r_i$  is the relative degree of the  $i$ -th output (relative degree  $r_i \leq 4$ , where  $r_i$  is the number of motors). Accordingly, the Taylor series is set to be lower than the number of active motors, as proposed, for example, in [27]. This proposed approach is implemented for a specific case study and compared with a conventional control approach. The obtained results demonstrate the engineering feasibility, robustness, and enhanced performance of the proposed approach. Note that four active motors are defined in the case of the study that is addressed in this paper. However, the proposed approach can be generalized to any CDPR.

The paper is organized into six sections. Section 2 is dedicated to the proposed robot modeling required for the control strategy design. Section 3 presents the control problem formulation and the corresponding solution. Then, the stability analysis is discussed in the following section. In Section 5, simulation results are presented. Finally, conclusions are given.

## 2. Robot Structure and Modeling

This paper focuses at a two-degree of freedom (DOF) planar cable-driven parallel robot, whose kinematic scheme and main geometric parameters are reported in Figure 1. This specific design was introduced in [28]. It is suitable for multiple applications, from conventional pick and place tasks to rehabilitation exercises of human upper limbs. This case study was selected since its dynamics are well known. However, the proposed control approach can be generalized, and it can be conveniently applied, for example, to spatial cable-driven robots.



**Figure 1.** (a) Planar four-cable-driven parallel robot structure. (b) The structure diagram of the pulley/shaft system.

The main geometric parameters of the robot are described in Table 1 by referring to the schemes in Figure 1.

**Table 1.** Nomenclature of the robot geometric parameters.

Parameters	Nomenclature
$A_i = (A_{ix}, A_{iy})$	Attachment points of the driving cables to the motorized pulleys.
$L_i$	Driven cables lengths
$\theta_i$	Cables angles with respect to the x-axis.
$L_B$	The lengths of the workspace side.
$X_m$	End-effector's x-coordinate.
$Y_m$	End-effector's y-coordinate.
$M$	End-effector's mass.
$T_i$	Motors torques.
$J_i$	Rotational inertia of shaft-pulley system.
$C_i$	Motors shafts viscous damping coefficients.
$\beta_i$	Pulleys' rotation angles.
$r_i$	Pulleys radius.

The dynamics of the platform are given by the following equation:

$$\mathbf{M}_{eq}(\mathbf{X})\ddot{\mathbf{X}} + \mathbf{N}(\mathbf{X}, \dot{\mathbf{X}}) = \mathbf{F}_v \quad (1)$$

where the bold type indicates a matrix or vector;  $\ddot{\mathbf{X}}$  is the inertial matrix;  $\mathbf{N}(\mathbf{X}, \dot{\mathbf{X}})$  is the matrix of Coriolis that can be written as

$$\mathbf{M}_{eq}(\mathbf{X}) = \mathbf{r} * \mathbf{m} + \mathbf{S}(\mathbf{X}) \mathbf{J} \frac{\partial \beta}{\partial \mathbf{x}}; \mathbf{N}(\mathbf{X}, \dot{\mathbf{X}}) = \mathbf{S}(\mathbf{X}) \left( \mathbf{J} \frac{d}{dt} \left( \frac{\partial \beta}{\partial \mathbf{x}} \right) + \mathbf{C} \frac{\partial \beta}{\partial \mathbf{x}} \right) \dot{\mathbf{X}} \quad (2)$$

where  $\mathbf{r}$  is the matrix of the pulley radiuses  $\mathbf{r}_i = \mathbf{r}$  and

$$\mathbf{m} = \begin{bmatrix} \mathbf{m} & \mathbf{0} \\ \mathbf{0} & \mathbf{m} \end{bmatrix}; \mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{J}_4 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} \mathbf{c}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{c}_4 \end{bmatrix}; \mathbf{X} = [\mathbf{x}_m \quad \mathbf{y}_m]^T \quad (3)$$

where the  $T$  apex represents the transpose operator.

The virtual Cartesian force input  $\mathbf{F}_V$  is defined as

$$\mathbf{F}_V = \mathbf{S}(\mathbf{X}) \boldsymbol{\tau} \quad (4)$$

The pulley angle vector can be written as a function of driven cable lengths as

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1(\mathbf{X}) \\ \vdots \\ \beta_4(\mathbf{X}) \end{bmatrix} = \frac{1}{\mathbf{r}} \begin{bmatrix} L_{01} - \sqrt{(x_m - A_{1x})^2 + (y_m - A_{1y})^2} \\ \vdots \\ L_{04} - \sqrt{(x_m - A_{4x})^2 + (y_m - A_{4y})^2} \end{bmatrix} \quad (5)$$

The Jacobian matrix  $\mathbf{S}(\mathbf{X})$  is given by

$$\mathbf{S}(\mathbf{X}) = \begin{bmatrix} -\cos \theta_1 & \cdots & -\cos \theta_4 \\ -\sin \theta_1 & \cdots & -\sin \theta_4 \end{bmatrix} \quad (6)$$

where

$$\theta_i = \text{atan}^{-1} \left( \frac{x_m - A_{ix}}{y_m - A_{iy}} \right) \quad (7)$$

where  $i = 1, \dots, 4$ .

Choosing  $\mathbf{x} = [\mathbf{x}_1 \quad \mathbf{x}_2]^T = [\mathbf{X}^T \quad \dot{\mathbf{X}}^T]^T$  as state vector and  $\mathbf{U} = \mathbf{F}_V$  as the control input, the proposed robot dynamic model given by Equation (1) can be rewritten in the form of

$$\begin{cases} \dot{\mathbf{x}}_1 = \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{U} \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) = \mathbf{x}_1 \end{cases} \quad (8)$$

where

$$\mathbf{f}(\mathbf{x}) = -\mathbf{N}(\mathbf{x}); \mathbf{g}(\mathbf{x}) = \mathbf{M}_{eq}^{-1}(\mathbf{x}) \quad (9)$$

### 3. The Proposed NCGCP Controller Design

One can consider the following multiple input–multiple output (MIMO) nonlinear system, whose nomenclature of parameters variables is shown in Table 2.

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) + \sum_{i=1}^m \mathbf{g}_i(\mathbf{x})u_i = \mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x}) \mathbf{U} \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) = [\mathbf{h}_1(\mathbf{x}), \mathbf{h}_2(\mathbf{x}), \dots, \mathbf{h}_m(\mathbf{x})]^T \end{cases} \quad (10)$$

where  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{U} \in \mathbb{R}^m$  and  $\mathbf{y} \in \mathbb{R}^m$  are, respectively, the state vector, the control vector and the output vector.  $\mathbf{F}(\mathbf{x})$  and  $\mathbf{h}(\mathbf{x})$  are smooth vector fields.

**Table 2.** Nomenclature of the multiple input–multiple output (MIMO) system parameters variables.

Parameters	Nomenclature
$\mathbf{x}$	the states vector
$\mathbf{U}$	control vector
$\mathbf{y}$	output vector
$\mathbf{F}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$	smooth vector fields
$\mathbf{y}(t + T)$	future reference
$\mathbf{e}(t + T)$	future tracking error vector
$\mathbf{y}_r(t + T)$	quadratic performance
$\prod_i^s$	the last row of the matrix
$\prod_i^{ss}$	is the last element of $\prod_i^s$
$\mathbf{K}_i$	is the $(\rho_i \times 1)$ gain vector
$\rho_i$	relative degree

As in the nonlinear case, the predictive control is based in taking the derivatives of the output. These output derivatives are obtained from the system of Equation (10). The proposed NCGPC control strategy allows us to minimize the error between the desired trajectory and the predicted positions while taking into account constraints on cable tensions. The NCGPC controller is designed so that the future output vector  $\mathbf{y}(t + T)$  follows the future reference signal  $\mathbf{y}_r(t + T)$  by minimizing the quadratic performance index given by

$$J = \int_0^T \mathbf{e}(t + \tau)^T \mathbf{e}(t + \tau) d\tau \quad (11)$$

where  $\mathbf{e}(t + T)$  is the future tracking error vector, and  $T > 0$  is the prediction horizon.

To solve the nonlinear optimization problem in Equation (11), the predicted term  $\mathbf{e}(t + T)$  is expanded into a  $(\rho + 1)^{\text{th}}$  order Taylor series expansion using the Lie derivative  $hm(\mathbf{x})$  along a field of vectors  $\mathbf{F}(\mathbf{x})$ . The outputs of the system and the reference signals predictions are approximated by their Taylor-series expansions up to corresponding relative degree  $\rho_i$  in the form

$$\mathbf{e}_i(t + T) \approx \sum_{k=0}^{\rho_i} \frac{T^k}{k!} \left( \mathbf{y}_{r_i}^{(k)}(t) - \mathbf{y}_i^{(k)}(t) \right) \quad (12)$$

$$\mathbf{e}_i(t + T) \approx \begin{bmatrix} 1 & T & \frac{T^2}{2!} & \cdots & \frac{T^{\rho_i}}{\rho_i!} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{r_i}(t) - \mathbf{y}_i(t) \\ \dot{\mathbf{y}}_{r_i}(t) - \dot{\mathbf{y}}_i(t) \\ \ddot{\mathbf{y}}_{r_i}(t) - \ddot{\mathbf{y}}_i(t) \\ \vdots \\ \mathbf{y}_{r_i}^{(\rho_i)}(t) - \mathbf{y}_i^{(\rho_i)}(t) \end{bmatrix} \quad (13)$$

$$\mathbf{e}_i(t + T) = \Gamma_i(T) \mathbf{Y}_i(t) \quad (14)$$

One can define the augmented vector  $\mathbf{Y}(t)$  as

$$\mathbf{Y}(t) = \begin{bmatrix} \mathbf{Y}_1^T(t) & \cdots & \mathbf{Y}_m^T(t) \end{bmatrix}^T \quad (15)$$

Therefore, one can write

$$\mathbf{e}(t + T) = \Gamma(T) \mathbf{Y}(t) \quad (16)$$

where  $\Gamma(T)$  is the  $m \times (\rho + m)$  ( $\rho = \sum_{i=1}^m \rho_i$ ) matrix defined as follows

$$\Gamma(T) = \text{Diag}(\Gamma_i(T)) \quad (17)$$

Substituting Equations (16) into (11) leads to the following criterion function

$$J \approx \int_0^T \mathbf{Y}^T(t) \Gamma^T(\tau) \Gamma(\tau) \mathbf{Y}(t) d\tau = \mathbf{Y}^T(t) \left( \int_0^T \Gamma^T(\tau) \Gamma(\tau) d\tau \right) \mathbf{Y}(t) \quad (18)$$

$$\mathbf{J} \approx \mathbf{Y}^T(t) \prod(T) \mathbf{Y}(t) \quad (19)$$

where

$$\prod(T) = \int_0^T \Gamma^T(\tau) \Gamma(\tau) d\tau = \text{Diag}(\prod_1(T), \prod_2(T), \dots, \prod_m(T)) \quad (20)$$

The output  $y_i$  time derivatives required in the approximated performance index are expressed as given below using Lie derivatives [28]. Numbered lists can be added as follows

$$\left\{ \begin{array}{l} \dot{y}_1 = L_F h_1(\mathbf{x}) \\ \ddot{y}_1 = L_F^2 h_1(\mathbf{x}) \\ \vdots \\ y_1^{(\rho_1)} = L_F^{(\rho_1)} h_1(\mathbf{x}) + L_{g_1} L_F^{(\rho_1-1)} h_1(\mathbf{x}) u_1 + \dots + L_{g_m} L_F^{(\rho_1-1)} h_1(\mathbf{x}) u_m \\ \dot{y}_2 = L_F h_2(\mathbf{x}) \\ \vdots \\ y_1^{(\rho_m)} = L_F^{(\rho_m)} h_m(\mathbf{x}) + L_{g_1} L_F^{(\rho_m-1)} h_m(\mathbf{x}) u_1 + \dots + L_{g_m} L_F^{(\rho_m-1)} h_m(\mathbf{x}) u_m \end{array} \right. \quad (21)$$

Introducing Equation (21) into the approximated performance index (19) yields

$$\mathbf{J} \approx (\mathbf{A}(\mathbf{x}) - \mathbf{B}(\mathbf{x}, \mathbf{U}))^T \prod(T) (\mathbf{A}(\mathbf{x}) - \mathbf{B}(\mathbf{x}, \mathbf{U})) \quad (22)$$

where

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} y_{r_1} - h_1(\mathbf{x}) \\ \dot{y}_{r_1} - L_F h_1(\mathbf{x}) \\ \vdots \\ y_{r_1}^{(\rho_1)} - L_F^{(\rho_1)} h_1(\mathbf{x}) \\ y_{r_2} - h_2(\mathbf{x}) \\ \vdots \\ y_{r_m}^{(\rho_m)} - L_F^{(\rho_m)} h_m(\mathbf{x}) \end{bmatrix} \quad (23)$$

$$\mathbf{B}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} 0_{\rho_1 \times 1} \\ L_{g_1} L_F^{(\rho_1-1)} h_1(\mathbf{x}) u_1 + \dots + L_{g_m} L_F^{(\rho_1-1)} h_1(\mathbf{x}) u_m \\ 0_{\rho_2 \times 1} \\ L_{g_1} L_F^{(\rho_2-1)} h_2(\mathbf{x}) u_1 + \dots + L_{g_m} L_F^{(\rho_2-1)} h_2(\mathbf{x}) u_m \\ \vdots \\ 0_{\rho_m \times 1} \\ L_{g_1} L_F^{(\rho_m-1)} h_m(\mathbf{x}) u_1 + \dots + L_{g_m} L_F^{(\rho_m-1)} h_m(\mathbf{x}) u_m \end{bmatrix} \quad (24)$$

Finally, the control law that minimizes the approximated performance index is obtained by solving the system as proposed in [29] in the form

$$\frac{d\mathbf{J}}{d\mathbf{U}} = \left( \frac{d(\mathbf{A}(\mathbf{x}) - \mathbf{B}(\mathbf{x}, \mathbf{U}))}{d\mathbf{U}} \right)^T \prod(T) (\mathbf{A}(\mathbf{x}) - \mathbf{B}(\mathbf{x}, \mathbf{U})) = 0_{(\rho+m) \times 1} \quad (25)$$

Consequently, the control law is given by

$$\mathbf{U} = \mathbf{D}^{-1}(\mathbf{x}) \text{Diag} \left( \left( \prod_i^{\text{ss}} \right)^{-1} \prod_i^{\text{s}} \right) \mathbf{A}(\mathbf{x}) = \mathbf{D}^{-1}(\mathbf{x}) \text{Diag}(\mathbf{K}_i) \mathbf{A}(\mathbf{x}) \quad (26)$$

where  $\mathbf{D}(\mathbf{x})$  is the  $(m \times m)$  decoupling matrix,  $\Pi_i^s$  is the last row of the matrix  $\Pi_i$ ,  $\Pi_i^{ss}$  is the last element of  $\Pi_i^s$  and  $\mathbf{K}_i$  is the  $(\rho_i \times 1)$  gain vector that can be written as

$$\mathbf{D}(\mathbf{x}) = \begin{bmatrix} L_{g_1} L_F^{(\rho_1-1)} h_1(\mathbf{x}) & L_{g_2} L_F^{(\rho_1-1)} h_1(\mathbf{x}) & \cdots & L_{g_m} L_F^{(\rho_1-1)} h_1(\mathbf{x}) \\ L_{g_1} L_F^{(\rho_2-1)} h_2(\mathbf{x}) & L_{g_1} L_F^{(\rho_2-1)} h_2(\mathbf{x}) & \cdots & L_{g_m} L_F^{(\rho_2-1)} h_2(\mathbf{x}) \\ \vdots & \vdots & \vdots & \vdots \\ L_{g_1} L_F^{(\rho_m-1)} h_m(\mathbf{x}) & L_{g_2} L_F^{(\rho_m-1)} h_m(\mathbf{x}) & \cdots & L_{g_m} L_F^{(\rho_m-1)} h_m(\mathbf{x}) \end{bmatrix} \quad (27)$$

$$\mathbf{K}_i = \left[ \frac{\rho_i!}{T^{\rho_i}} \frac{2\rho_i+1}{(\rho_i+1)} \quad \cdots \quad \frac{\rho_i!}{T^{\rho_i-j}} \frac{2\rho_i+1}{j!(\rho_i+j+1)} \quad \cdots \quad 1 \right] \quad (28)$$

#### 4. Stability Analysis

Under the assumption that all of the system's zero dynamics are asymptotically stable, proving the stability of the closed-loop system leads to proving that the output tracking errors converge to zero. Replacing the control law given by Equation (26) into (21) yields the following dynamic tracking errors:

$$e_i^{(\rho_i)}(t) + K_{i(\rho_i-1)} e_i^{(\rho_i-1)}(t) + \cdots + K_{i1} \dot{e}_i(t) + K_{i0} e_i(t) = 0 \quad (29)$$

As given in [15], for relative degrees with  $\rho_i \leq 4$ , closed-loop system stability is guaranteed. However, instability occurs when the relative degree is greater than four. For the proposed robot, we have

$$\left\{ \begin{array}{l} \rho_1 = \rho_2 = 2 \\ \mathbf{D}(\mathbf{x}) = \mathbf{M}_{eq}(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) = \mathbf{X} = \begin{bmatrix} x_m & y_m \end{bmatrix}^T \\ L_f \mathbf{h}(\mathbf{x}) = \begin{bmatrix} L_f h_1(\mathbf{x}) & L_f h_2(\mathbf{x}) \end{bmatrix}^T = \dot{\mathbf{X}} \\ L_f^2 \mathbf{h}(\mathbf{x}) = \begin{bmatrix} L_f^2 h_1(\mathbf{x}) & L_f^2 h_2(\mathbf{x}) \end{bmatrix}^T = -\mathbf{N}(\mathbf{x}) \\ \mathbf{K}_1 = \mathbf{K}_2 = \begin{bmatrix} \frac{10}{3T^2} & \frac{10}{4T} & 1 \end{bmatrix} \end{array} \right. \quad (30)$$

Therefore, the control that minimizes the performance index (11) is given by

$$\mathbf{U} = \mathbf{M}_{eq}(\mathbf{x}) \left[ \begin{array}{l} \sum_{i=0}^{\rho_1-2} K_{1i} (y_{r1}^{(i)}(t) - L_f^i h_1(\mathbf{x})) \\ \sum_{i=0}^{\rho_2-2} K_{2i} (y_{r2}^{(i)}(t) - L_f^i h_2(\mathbf{x})) \end{array} \right] \quad (31)$$

This leads to the closed-loop tracking error dynamics given by

$$\left\{ \begin{array}{l} \ddot{e}_1(t) + K_{11} \dot{e}_1(t) + K_{10} e_1(t) = 0 \\ \ddot{e}_2(t) + K_{21} \dot{e}_2(t) + K_{20} e_2(t) = 0 \end{array} \right. \quad (32)$$

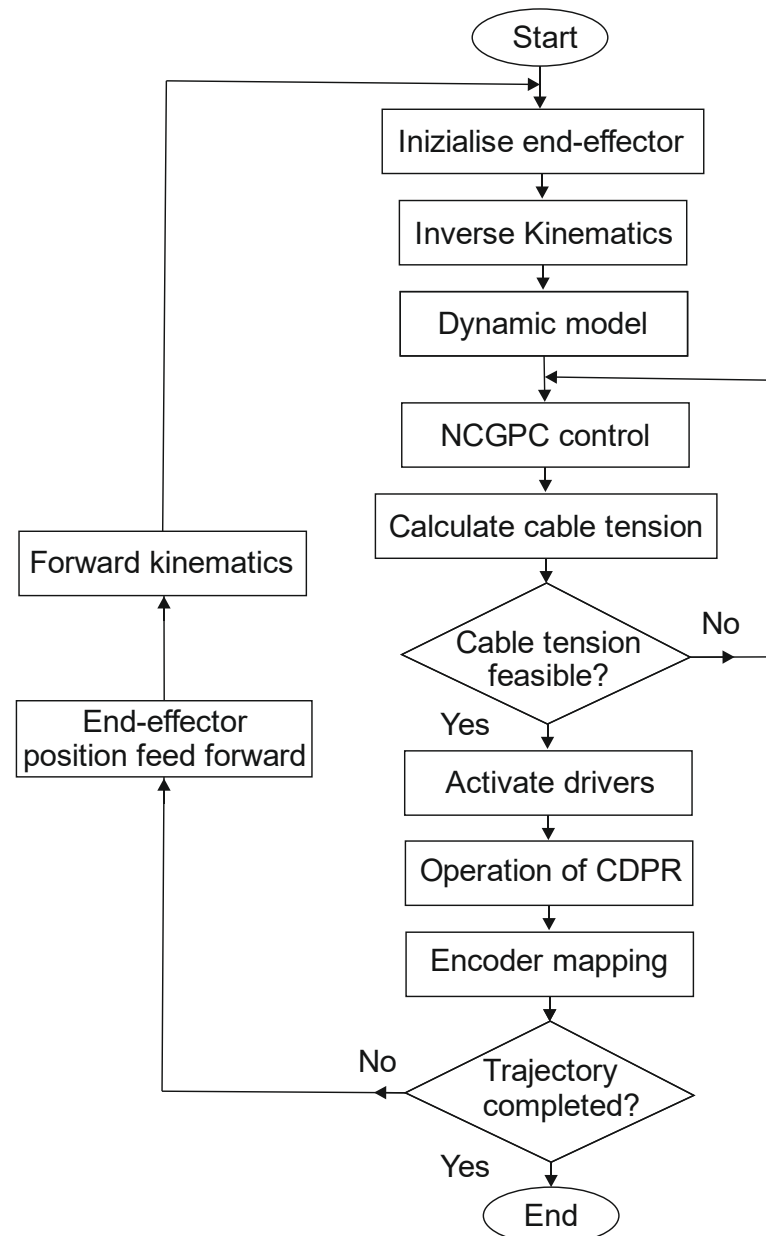
Error dynamics correspond to the following closed-loop eigenvalues

$$\left\{ \begin{array}{l} \lambda_{1,2} = -\frac{1.25}{T} \pm j \frac{1.33}{T} \\ \lambda_{3,4} = -\frac{1.25}{T} \pm j \frac{1.33}{T} \end{array} \right. \quad (33)$$

Therefore, stability is proven for the proposed CDPR model, and the position error between the actual position and the desired position is real-time feed forward.

The proposed formulation uses the reported inverse kinematics equations. Given the Cartesian position  $\mathbf{X}_R = \{x_R \ y_R\}^T$ , we calculated the cable lengths  $L_i$  at a given time. Given the specific kinematic formulation, the proposed formulation can be easily obtained from the inverse kinematics without requiring a close-form solution of the direct kinematics. Figure 2 shows a flowchart of the entire control scheme. After the desired control target is

provided, the robot must initialize its end-effector position by solving the inverse position kinematics problem; that is, it is given the Cartesian position  $\mathbf{X}_R = \{x_R, y_R\}^T$ , where  $\mathbf{X}_R$  is the input of the reference trajectory. Then, one can obtain the corresponding cable lengths  $L_i$ . The solution is based on calculating the Euclidean norm between the moving point  $\mathbf{X}_R = \{x_R, y_R\}^T$  and each fixed ground link vertex  $A_i$  where the cables originate. The dynamics model and the control technique calculate the cable tensions with a limit to the feasible cable tensions (they must always be positive), as also mentioned in [30–35]. Then, the system is driven to generate the trajectory.



**Figure 2.** A flow-chart of tension control (NCGPC: nonlinear continuous time generalized predictive control; PDP robots: planar-driven parallel robots).

The proposed control architecture includes a feedforward term, and a virtual Cartesian force input to real actuator torque calculation. The PD controller gains are determined via trial-and-error using a Matlab Simulink simulation to achieve reasonable performance. The matrix gains  $\mathbf{K}_P$  and  $\mathbf{K}_D$  are  $2 \times 2$  diagonal matrices, which means that the PD control is accomplished independently for the x and y motions, even though the dynamics model is



coupled. The feedforward term  $\mathbf{M}_{eq}(\mathbf{X})$  is composed of the overall position-dependent Cartesian mass matrix and the reference Cartesian acceleration components. One should note that the virtual-to-real calculation has the task of inverting the matrix  $\mathbf{S}(\mathbf{X})$  that is nonsquare with the constraint of only positive cable tensions.

## 5. Simulation Results

In this section, simulation results obtained with NCGPC are presented and compared to those obtained using computed torque control (CTC), which is widely used for robotic systems. Both simulations were carried out within Matlab environment.

The CTC controller is defined as follows

$$\mathbf{U} = \mathbf{M}_{eq}(\mathbf{X}) [\ddot{\mathbf{X}}_r + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e}] + \mathbf{N}(\mathbf{X}, \dot{\mathbf{X}}) \quad (34)$$

where

$$\mathbf{e} = \mathbf{X}_r - \mathbf{X} \quad (35)$$

$\mathbf{K}_P$  and  $\mathbf{K}_D$  are  $2 \times 2$  diagonal matrices containing the linear gains. Applying this control to the robot model (1) yields the tracking errors dynamics as

$$\ddot{\mathbf{e}} + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} = 0 \quad (36)$$

Based on the above-mentioned formulations, both NCGPC and CTC control schemes were implemented based on the control scheme in Figure 3.

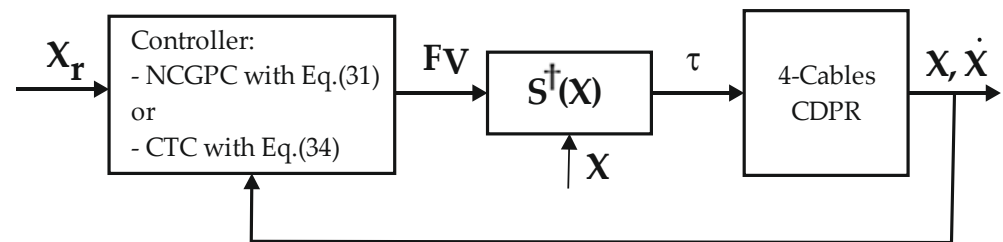


Figure 3. Proposed control schemes for the proposed planar 4-cable CDPR.

The parameters for the dynamic equations of motion (34) for the 4-cable CDDRs are: mass  $m = 0.01\text{--}0.015$  kg; rotational shaft/pulley inertias  $J_i$  ( $i = 1, \dots, 4$ ) =  $0.0008$  kg  $\text{m}^2$ . Shaft rotational viscous damping coefficients  $C_i$  ( $i = 1, \dots, 4$ ) were set as equal to  $0.01$  Nms and  $r_i = r = 1$  cm (for all  $i = 1, \dots, 4$ ) by considering the properties of standard off-the-shelf components. We considered the reference of our system in the center of the workspace  $(0,0)$ . Accordingly, we wrote the nonlinear dynamics terms  $\mathbf{N}(\mathbf{X}, \dot{\mathbf{X}})$  in Equation (34). Then, we set the reference Cartesian acceleration  $\ddot{\mathbf{X}}_r$  in a feed forward term, and we implemented a Cartesian PD controller to reduce the tracking error  $\mathbf{e} = \mathbf{X}_r - \mathbf{X}$ . The reference Cartesian position is given by  $\mathbf{X}_R = \{x_R \ y_R\}^T$ .

$\mathbf{S}^\dagger(\mathbf{X})$  is the pseudo-inverse of  $\mathbf{S}(\mathbf{X})$ . The reference trajectory is chosen as a circular path given by

$$\begin{cases} x_{mref} = 0.2165 * \sin(\alpha(t)) \\ y_{mref} = 0.2165 * \cos(\alpha(t)) \end{cases} \quad (37)$$

where  $\ddot{\alpha}(t) = 8\pi$ ,  $\alpha(0) = \dot{\alpha}(0) = 0$ .

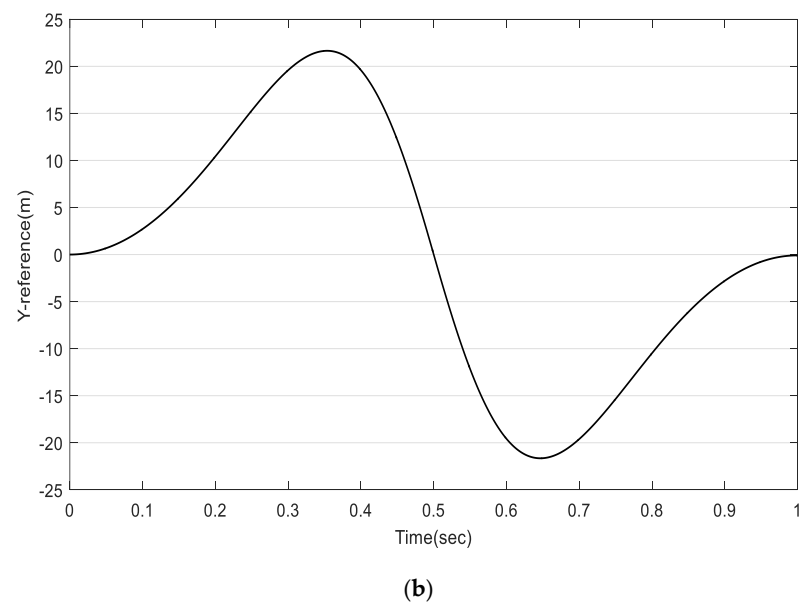
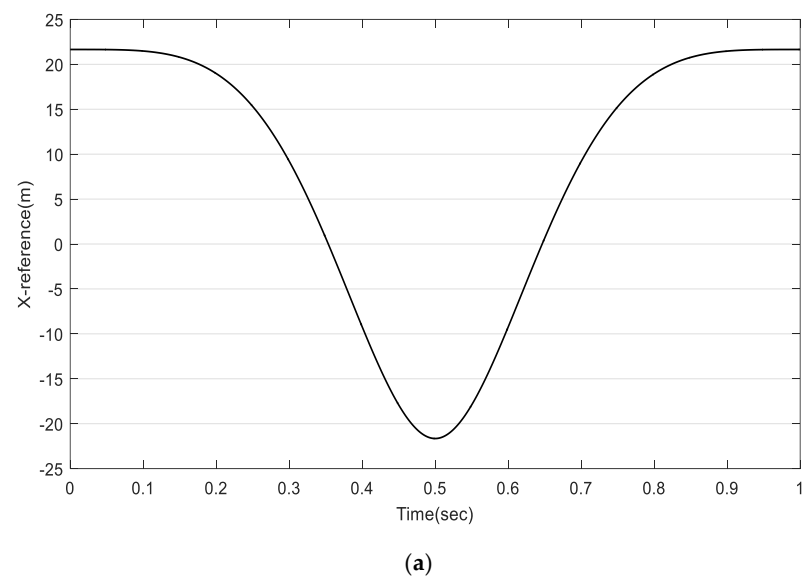
Referring to Table 1, the parameters for the dynamic Equation (34) for both the 4-cable CDDRs are assumed as mass  $m = 0.01$  kg and rotational shaft/pulley inertias  $J_i$  ( $i = 1, \dots, 4$ ) =  $0.0008$  kg  $\text{m}^2$ . Shaft rotational viscous damping coefficients  $C_i$  ( $i = 1, \dots, 4$ ) =  $0.01$  Nm s and  $r_i = r = 1$  cm (for all  $i = 1, \dots, 4$ ), and the base square side is  $L_b = 0.658$  m, as also summarized in Table 3. We defined the reference frame of our CDPR system in the center of the workspace with coordinates  $(0,0)$ .

**Table 3.** Main values of parameters for the proposed case study.

Parameters	Value	Parameters	Value
$\begin{pmatrix} A_{1x}, A_{1y} \end{pmatrix}$	(−32.5;0) cm	$L_B$	65 cm
$\begin{pmatrix} A_{1x}, A_{1y} \end{pmatrix}$	(32.5;0) cm	$M$	0.01 kg
$\begin{pmatrix} A_{1x}, A_{1y} \end{pmatrix}$	(32.5;32.5) cm	$J_i$	0.0008 kg m <sup>2</sup>
$\begin{pmatrix} A_{1x}, A_{1y} \end{pmatrix}$	(−32.5;32.5) cm	$C_i$	0.01 Nms

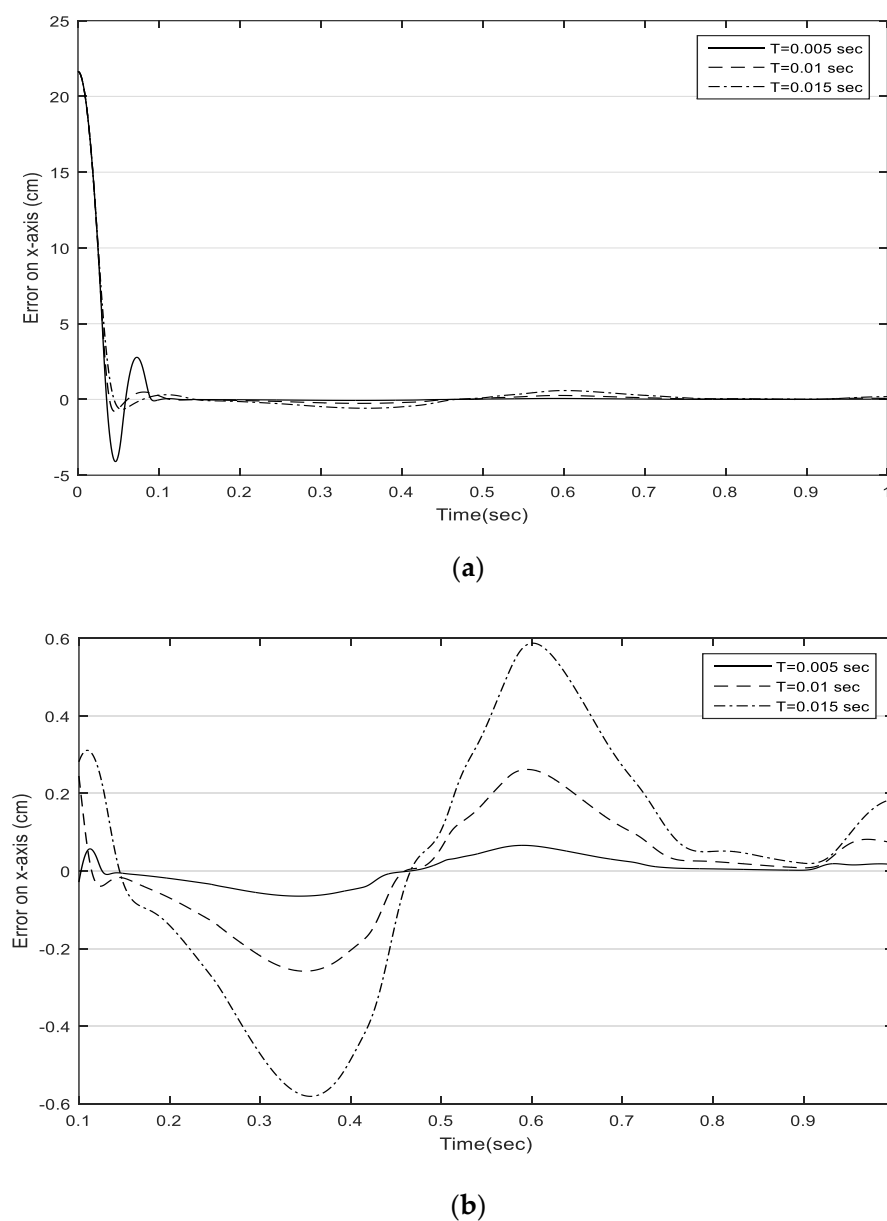
Figure 4 shows the plots of the reference trajectory for X- and Y-coordinates. The proposed simulation study was carried out with three phases:

- Phase 1: Influence of the prediction horizon on NCGPC tracking performances.
- Phase 2: Comparison between NCGPC and CTC based on tracking errors.
- Phase 3: Robustness test of payload and parameter uncertainties for both NCGPC and CTC.

**Figure 4.** A plot of the reference trajectory: (a) X-coordinate; (b) Y-coordinate.

### 5.1. Influence of the Prediction Horizon on NCGPC Tracking Performances

Figure 5 shows tracking errors on the X-axis for different prediction horizons and a zoom on X-axis tracking errors (b). Figures 5 and 6 show the tracking performances that can be achieved by the proposed NCGPC control with a maximum error lower than 0.6 cm on the X-axis and 0.9 cm on the Y-axis in the worst case at  $t = 0.015$  s. The prediction horizon exerts a direct influence on the transient response because it acts directly on the pole placement of the closed loop. In addition, the steady state error is affected by the prediction horizon, which is due to the precision decrease in the prediction model when the prediction horizon becomes longer. In Figure 7, motor torques are shown. The lower and upper bounds are limited (between 0.01–10 Nm) to avoid negative and high tensions, which can lead to a cable loosening or tearing. Figure 8 shows resulting tensions in robot cables. Cable tensions are positive, except in the transient response, where they are slightly negative.



**Figure 5.** (a) NGPC tracking errors on X-axis for different prediction horizons. (b) Zoom on X-axis tracking errors.

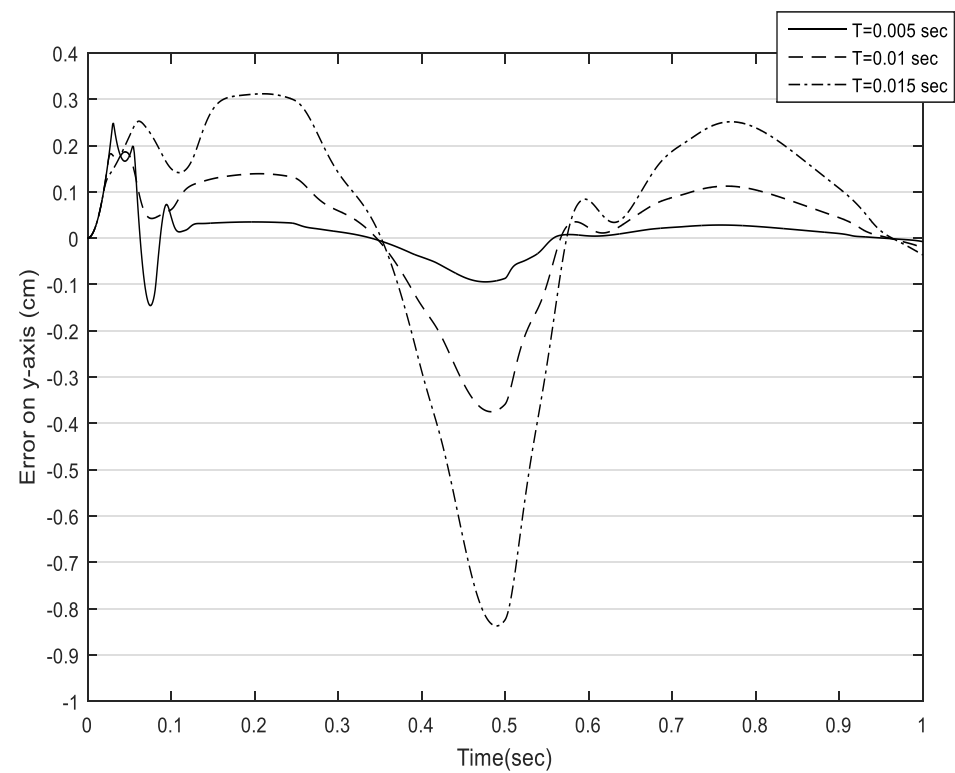


Figure 6. NGPC tracking errors on Y-axis for different prediction horizons.

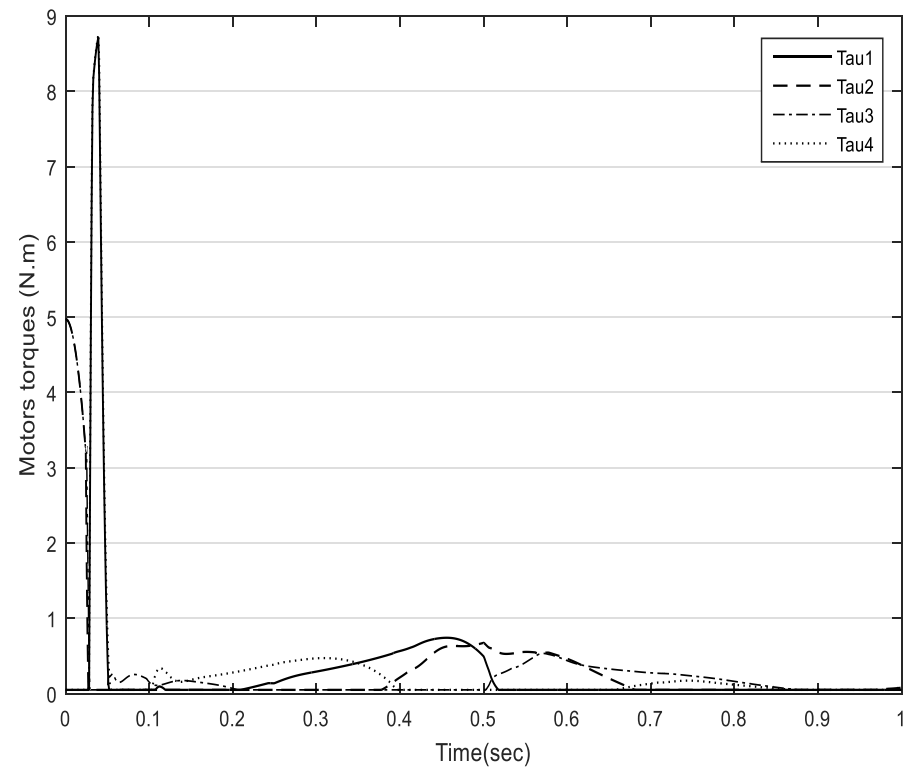
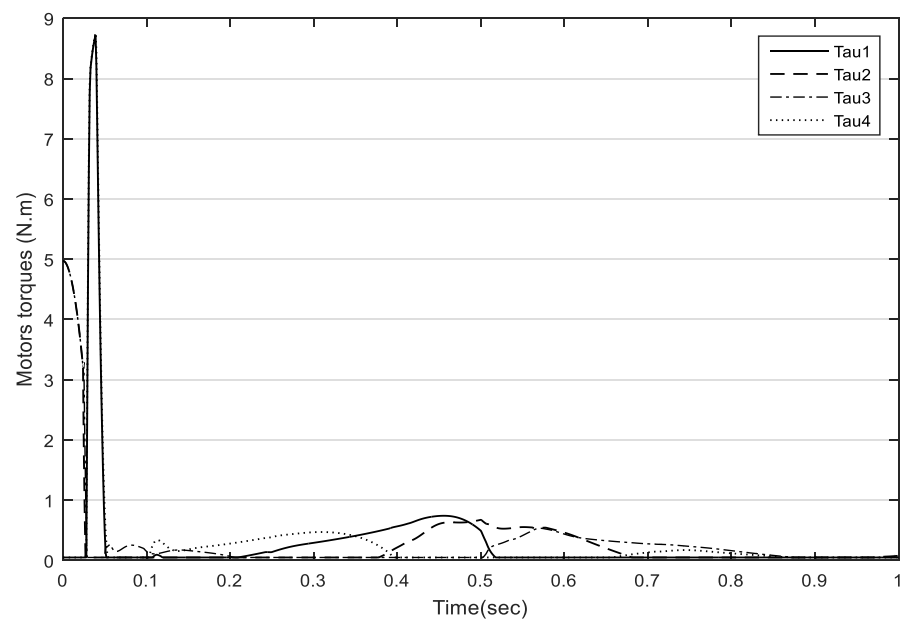
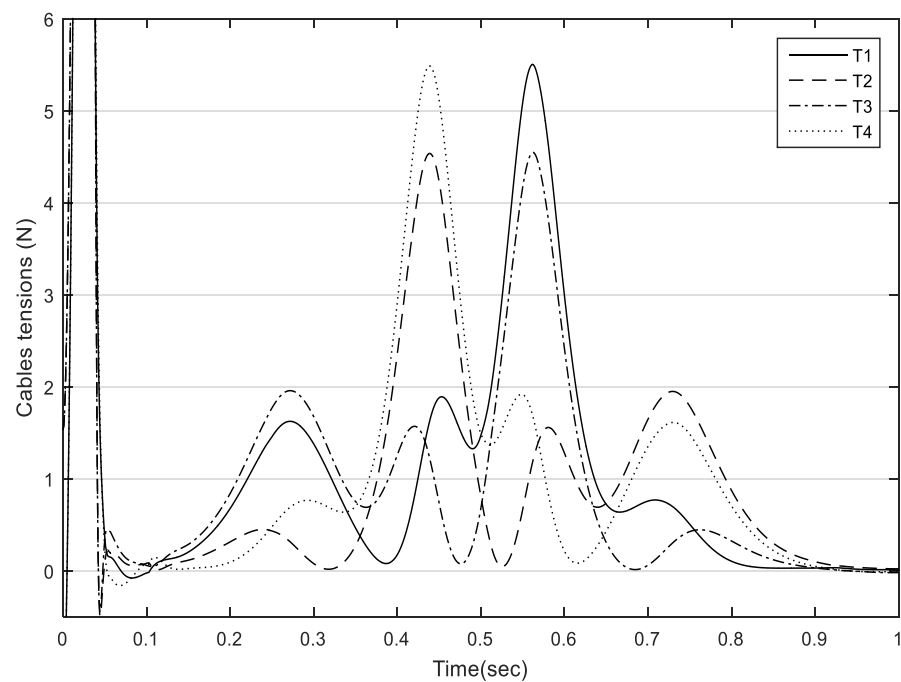


Figure 7. Motors torques produced by NGPC for  $T = 0.01$  s.



(a)



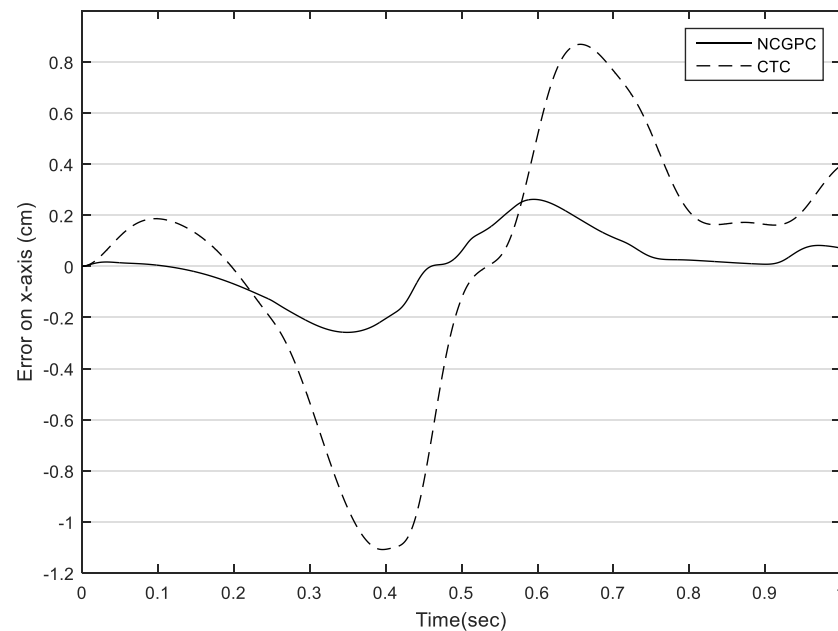
(b)

**Figure 8.** (a) Cables tensions. (b) Zoom on cable tensions.

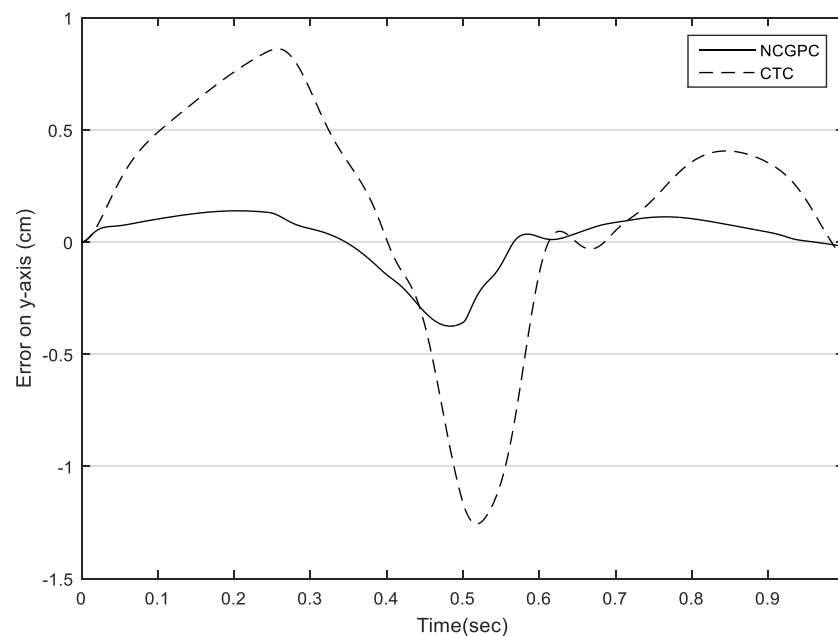
### 5.2. Comparison between NCGPC and CTC Based on Tracking Errors

Figures 9 and 10 show the tracking errors along X- and Y-directions of the end-effector under different control strategies. In order to illustrate the superiority of the NCGPC scheme, we compared it with CTC. In particular, one can identify a positioning error ranging from +0.2 to +0.4 cm for the NCGPC control method, while one can identify a positioning error ranging from −1.3 to +0.8 cm for the CTC control. One can also compute the least square average positioning error along the X-axis as 0.04 cm with NCGPC and 0.2 cm with CTC when referring to the interval from 0.0 to 0.3 s. The least square average positioning error along the X-axis is 0.2 cm with NCGPC and 0.8 cm with CTC when

referring to the interval from 0.5 to 0.8 s. Similarly, the least square average positioning error along the X-axis is 0.2 cm with NCGPC and 0.8 cm with CTC when referring to the interval from 0.0 to 0.3 s. The least square average positioning error along the X-axis is 0.1 cm with NCGPC and 0.4 cm with CTC when referring to the interval from 0.5 to 0.8 s.



**Figure 9.** Position tracking errors along X-direction.



**Figure 10.** Position tracking errors along Y-direction.

As seen for these values, we can conclude that the NCGPC has a higher position tracking accuracy compared with other control methods with different prediction horizons. One should note that NCGPCs are extended from affine nonlinear systems to general nonlinear systems and strategies applied to a permanent magnet motor for speed trajectory tracking and disturbance rejection. Furthermore, they are revised in order to enhance the robustness of the closed-loop system. Computed torque control (CTC) is instead an

effective and high-performance motion control strategy for a robotic system. Based on the simulation error results, we can conclude that the proposed control method is effective and robust and can outperform CTC.

### 5.3. Robustness Test of Payload and Parameter Uncertainties

Measurement uncertainty is a nonnegative parameter characterizing the dispersion of the quantity values attributed to a measure and based on the information used. The guide to the expression of uncertainty in measurement (GUM) is the internationally accepted reference for evaluating uncertainty, and it requires the use of a first-order Taylor series expansion for propagating uncertainties. The specific assumed uncertainty values are reported in Table 4.

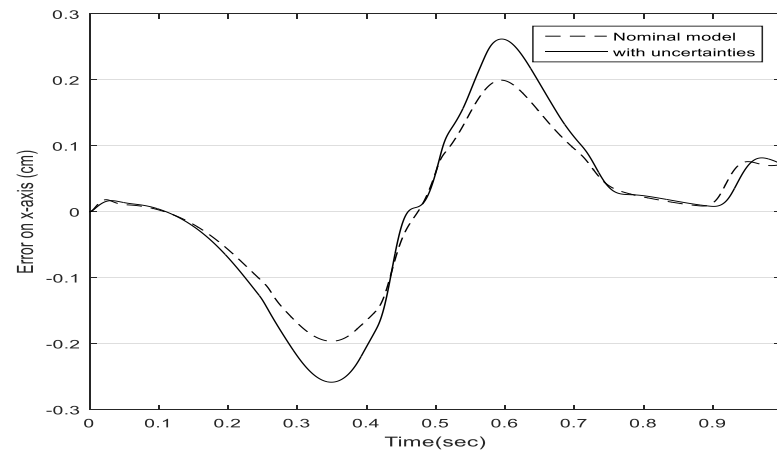
**Table 4.** List of the considered parameter uncertainties.

Parameters	Uncertainties
$\Delta r$	+5%
$\Delta J$	−10%
$\Delta L_b$	−5%
$\Delta c$	+10%
Payload: $\Delta m$	+20%

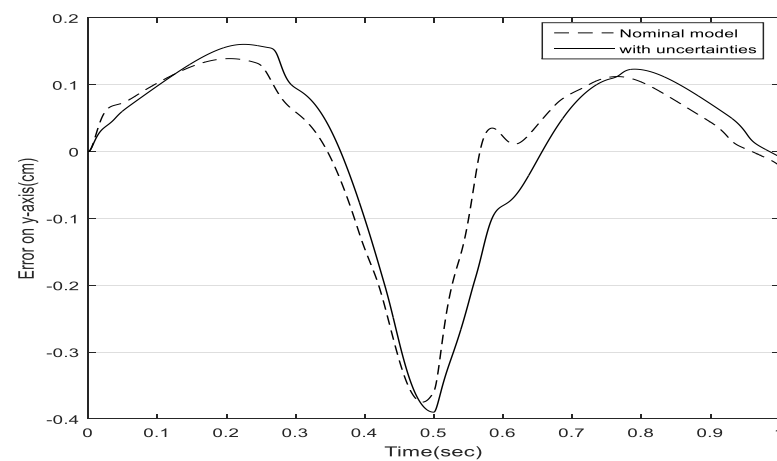
The introduced uncertainties, which were implemented in numerical modeling with Matlab Simulink software, are given in Table 4. Using the proposed robust control toolbox, one can analyze the effect of model uncertainty on the closed-loop stability and performance of the control system. In particular, one can determine whether our controller will perform adequately over the entire considered operating range, and which source of uncertainty is most likely to jeopardize the performance. The considered uncertainties are mostly related to the cable lengths. Cables can only pull, so there are only feasible positive cable tensions. The elasticity of cables makes them elongate proportionally to cable tensions. Accordingly, only positive increases in the cable lengths can be feasible under positive cable tensions as compared with nominal values. Although a sensitivity analysis is out of the scope of this work, the proposed formulation allows for tolerances in the setup parameters commonly above 1% of full scale, as reported in the plots in Figures 11–14.

We noticed that the position versus time curves obtained with the NCGPC method with the uncertainties follow (superpose to) the curve of the nominal model in the interval from 0.0 to 0.2 s very well. Similarly, in the interval from 0.22 to 0.4 s, we observe that the model with uncertainties follows the curve of the nominal NCGPC control model very well, with a similar tracking error ranging between 0.03 to 0.05 cm. In the interval from 0.5 to 1 s, the curve of NCGPC with uncertainties follows the curve of the nominal model with a maximum tracking error of 0.04 cm along both the X- and Y- directions. On the other hand, the position versus time curves obtained with the CTC method when considering the same uncertainties demonstrate a significantly different behavior; that is, the interval from 0.0 to 2.0 s shows a similar tracking performance but starting from the interval from 0.02 to 0.1, the curve with uncertainties demonstrates a significantly worst tracking error performance with a large tracking error between 0.03 and 1 cm along both the X- and Y-directions. When compared to Figures 9 and 10, the position tracking errors along the X-direction with the NCGPC method show a tracking error performance that is 98% better than the CTC in the interval from 0 to 2 s, where the tracking error results are negligible for NCGPC, while it is 0.2 cm with the CTC method. Similarly, the NCGPC control method shows a performance improvement as compared with the CTC method of about 90% in several specific time intervals. Considering absolute values, for example, NCGPC shows a tracking error of −0.23 cm, while CTC shows a tracking error of 1.16 cm at the time 4 s. Similarly, along the Y-direction at time 0.25 s, NCGPC shows a tracking error of 0.13 cm, while CTC shows a tracking error of 0.94 cm, again with NCGPC outperforming CTC by about 90%. Accordingly, one can outline that, when introducing model uncertainties

and payload mass, NCGPC demonstrates a significant improvement in tracking error performance as compared with a standard CTC, also in the presence of significant amounts of uncertainties.

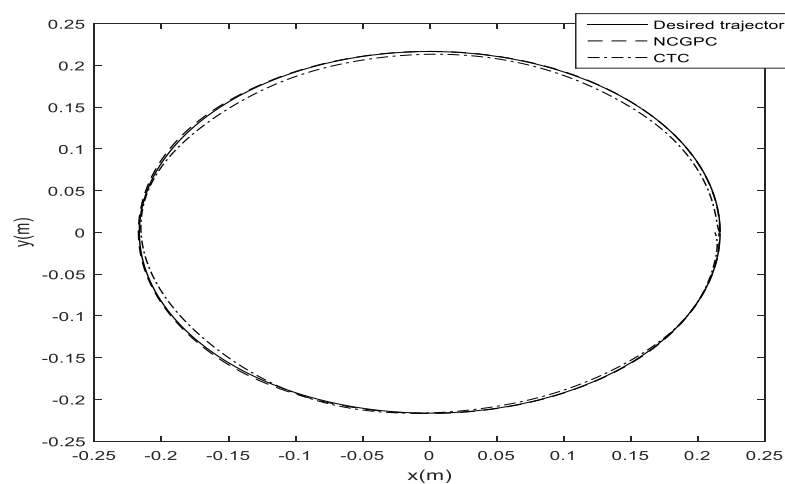


(a)



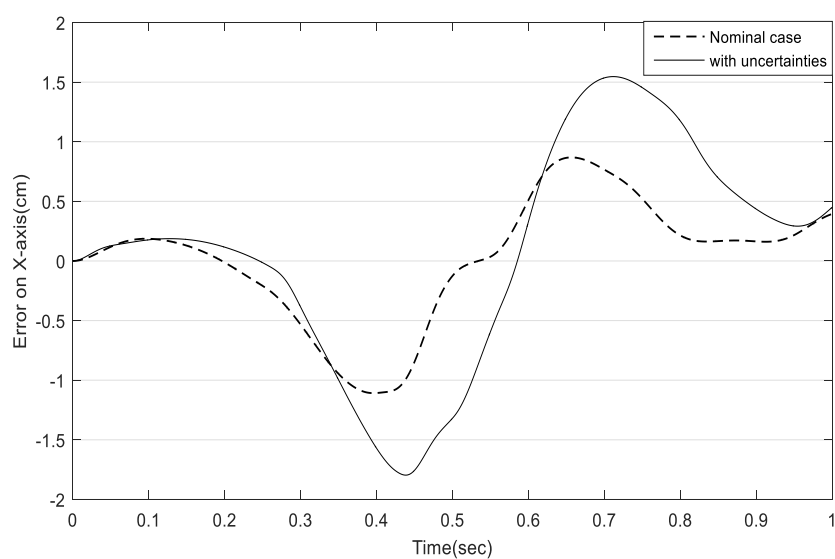
(b)

**Figure 11.** Effect of model uncertainties on NCGPC tracking errors: (a) X-axis; (b) Y-axis.

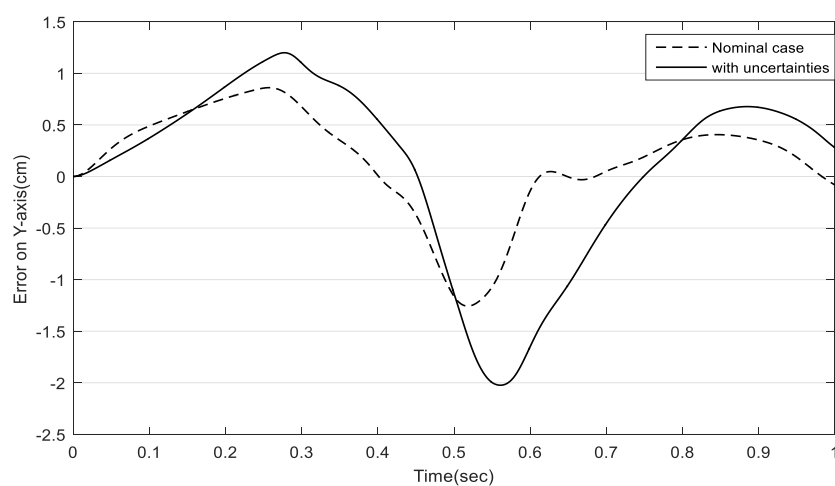


**Figure 12.** Desired and actual trajectories in task space.



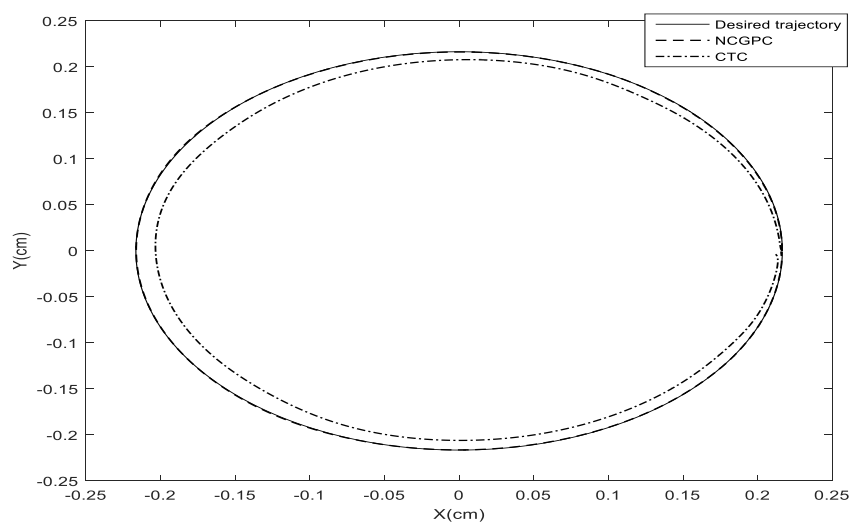


(a)



(b)

**Figure 13.** Effect of model uncertainties on CTC tracking errors: (a) X-axis; (b) Y-axis.



**Figure 14.** Effect of model uncertainties on both CTC and NCGPC tracking performance.

In this paper, the existing results for the analytic NCGPC are extended from affine nonlinear systems to general nonlinear systems (the proposed CDPR robot). This controller overcomes the difficulties imposed by the on-line nonlinear optimization involved in NNCGPC, such as attaining local-minimum and long-sampling-time requirements. The NCGPC developed in this paper is suboptimal for the original NCGPC problem. Since there is no restriction on the order for Taylor-series expansion, theoretically, the analytic solution developed by approximation can approach the solution of the original NCGPC at any specified accuracy.

This study is mostly focused on a specific planar CDPR. In future work, we plan to extend this formulation to a 3D CDPR, such as LAWEX and CUBE, with 3 to 6 active degrees of freedom; to reconfigurable cable-driven devices; and/or to devices with new application tasks, such as those reported in [36–41]. Additionally, multiobjective optimization strategies could be implemented at design and control stages, as proposed, for example, in [42,43].

## 6. Conclusions

This paper describes a novel continuous time predictive controller design method for a planar cable-driven parallel robot. The proposed method is based on the minimization of a quadratic cost function of the predicted tracking error as calculated using Taylor expansion. The main advantages of this proposed control scheme consist of avoiding on-line optimizations and the capability to deal with fast dynamics on nonlinear systems. Simulation results showed that the proposed NCGPC technique has a significantly improved tracking performance with respect to a standard computed torque control (CTC). Furthermore, the proposed control method demonstrates a suitable robustness in terms of the rejection of model uncertainties and payload variations, as also demonstrated in comparison with a CTC controller.

**Author Contributions:** Conceptualization, F.I. and G.C.; software, F.I. and A.M.; writing—original draft preparation, F.I. and A.M.; writing—review and editing, G.C., supervision, G.C.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Albus, J.; Bostelman, R.; Dagalakis, N. The NIST RoBoCrane. *J. Res. Natl. Inst. Stand. Technol.* **1992**, *94*, 373–384. [[CrossRef](#)] [[PubMed](#)]
2. Tang, X. An Overview of the Development for Cable-Driven Parallel Manipulator. *Adv. Mech. Eng.* **2014**, *6*, 823028.
3. Pinto, A.M.; Moreira, E.; Lima, J.; Sousa, J.P.; Costa, P. A cable-driven robot for architectural constructions: A visual-guided approach for motion control and path-planning. *Auton. Robots* **2017**, *41*, 1487–1499. [[CrossRef](#)]
4. Morizono, T.; Kurahashi, K.; Kawamura, S. Realization of a Virtual Sports Training System with Parallel Wire Mechanism. In Proceedings of the International Conference on Robotics and Automation, Albuquerque, NM, USA, 21–27 April 1997; Volume 4, pp. 3025–3030.
5. Reddy, M.V.; Praneet, N.C.; Ananthasuresh, G.K. Planar Cable-Driven Robots with Enhanced Orientability. In *Cable-Driven Parallel Robots Mechanisms and Machine Science* 74; Springer Nature: Cham, Switzerland, 2019; pp. 3–12.
6. Korayem, M.; Yousefzadeh, M.; Tourajizadeh, H. Optimal Control of a Wheeled Mobile Cable-Driven Parallel Robot ICaSbot with Viscoelastic Cables. *Robotica* **2020**, *38*, 1513–1537. [[CrossRef](#)]
7. Wei, H.; Qiu, Y.; Sheng, Y. On the Cable Pseudo-Drag Problem of Cable-Driven Parallel Camera Robots at High Speeds. *Robotica* **2019**, *37*, 1695–1709.
8. Hwang, S.; Bak, J.; Yoon, J.; Park, J. Oscillation Reduction and Frequency Analysis of Under-Constrained Cable-Driven Parallel Robot with Three Cables. *Robotica* **2020**, *38*, 375–395. [[CrossRef](#)]
9. Laroche, E.; Chellal, R.; Cuvillon, L.; Gangloff, J. A preliminary study for  $H_\infty$  control of parallel cable-driven manipulators. *Cable Driven Parallel Robot. Mech. Mach. Sci.* **2013**, *12*, 353–369.

10. Khosravi, M.A.; Taghirad, H.D. Dynamic analysis and control of cable driven robots with elastic cables. *Trans. Can. Soc. Mech. Eng.* **2011**, *35*, 543–558. [\[CrossRef\]](#)
11. Korayem, M.H.; Tourajizadeh, H.; Taherifar, M.; Korayem, A.H. Optimal feedback linearization control of a flexible cable robot. *Lat. Am. Appl. Res.* **2014**, *44*, 259–265.
12. Li, B.; Yan, L.; Gerada, C. The Novel Singular-Perturbation-Based Adaptive Control with  $\sigma$ -Modification for Cable Driven System. *Actuators* **2021**, *10*, 45. [\[CrossRef\]](#)
13. Fernández, J.D.; Yu, B.; Bargsten, V.; Zipper, M.; Sprengel, H. Design, Modelling and Control of Novel Series-Elastic Actuators for Industrial Robots. *Actuators* **2020**, *9*, 6.
14. Deng, E.; Tadesse, Y. A Soft 3D-Printed Robotic Hand Actuated by Coiled SMA. *Actuators* **2021**, *10*, 6. [\[CrossRef\]](#)
15. Laribi, A.; Carbone, G.; Zeghloul, S. On the Optimal Design of Cable Driven Parallel Robot with a Prescribed Workspace for Upper Limb Rehabilitation Tasks. *J. Bionic Eng.* **2019**, *16*, 503–513. [\[CrossRef\]](#)
16. Inel, F.; Menssouri, Z.; Carbone, G.; Ceccarelli, M. Dynamic Modeling and Simulation of Sliding Mode Control for a Cable Driven Parallel Robot. In *New Advances in Mechanism and Machine Science*; Springer: Cham, Switzerland, 2018; Volume 57, pp. 413–426.
17. Carbone, G.; Gherman, B.; Ulinici, I.; Vaida, C.; Pislă, D. Design issues for an inherently safe robotic rehabilitation device. In *New Advances in Mechanism and Machine Science*; Springer: Cham, Switzerland, 2018; Volume 49, pp. 1025–1032.
18. Tang, X.Q.; Shao, Z.F. Trajectory generation and tracking control of a multi-level hybrid support manipulator in FAST. *Mechatronics* **2013**, *23*, 1113–1122. [\[CrossRef\]](#)
19. Li, M.; Wu, H.; Wang, Y.; Handroos, H.; Carbone, G. Modified Levenberg-Marquardt algorithm for backpropagation neural network training in dynamic model identification of mechanical systems. *J. Dyn. Syst. Meas. Control Trans. ASME* **2017**, *139*. [\[CrossRef\]](#)
20. Poignet, P.; Gautier, M. Nonlinear model predictive control of a robot manipulator. In Proceedings of the 6th International Workshop on Advanced Motion Control. Proceedings (Cat. No.00TH8494), Nagoya, Japan, 30 March–1 April 2000; pp. 401–406. [\[CrossRef\]](#)
21. Chen, W.H. Predictive Control of General Non-linear Systems Using Approximation. *IEEE Proc. Control Theory Appl.* **2004**, *151*, 137–144. [\[CrossRef\]](#)
22. Hedjar, R.; Toumi, R.; Boucher, P.; Dumur, D. Finite Horizon Non-linear Predictive Control by the Taylor Approximation Application to Robot Tracking Trajectory. *Int. J. Appl. Math. Comput. Sci.* **2005**, *15*, 527–540.
23. Zhao, L.; Li, Q.; Liu, B.; Cheng, H. Trajectory tracking control of a one degree of freedom manipulator based on a switched sliding mode controller with a novel extended state observer framework. *IEEE Trans. Syst. Man Cybernet. Syst.* **2019**, *49*, 1110–1118. [\[CrossRef\]](#)
24. Dabo, M.; Chafouk, H.; Langlois, N. Unconstrained NCGPC with a Guaranteed Closed-Loop Stability: Case of Non-linear SISO Systems with the Relative Degree Greater than Four. In Proceedings of the 48th IEEE Conference on Decision and Control, Shanghai, China, 15–18 December 2009; pp. 1980–1985.
25. Siller-Alcalá, I.I.; Liceaga-Castro, J.U. Continuous Time Generalized Predictive Control from a Classical Control Perspective. *WSEAS Trans. Syst. Control* **2018**, *13*, 447–452.
26. Carbone, G.; Aróstegui Cervero, C.; Ceccarelli, M.; Altuzarra, O. A study of feasibility for a limb exercising device. In *New Advances in Mechanism and Machine Science*; Springer: Cham, Switzerland, 2018; Volume 47, pp. 11–21.
27. Errouissi, R.; Ouhrouche, M.; Chen, W.-H. Robust Non-linear Generalized Predictive Control of a Permanent Magnet Synchronous Motor with an Anti-Windup Compensator. In Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE), Bari, Italy, 4–7 July 2010; pp. 3184–3189.
28. Krid, M.; Benamar, F.; Lenain, R. A New Explicit Dynamic Path Tracking Controller Using Generalized Predictive Control. *Int. J. Control Automat. Syst.* **2017**, *15*, 303–314.
29. Siller-Alcalá, I.I.; Liceaga-Castro, J.U.; Alcántara-Ramírez, R.; Jaimes-Ponce, J. Non-linear Control of a Single-Link Flexible Joint Manipulator via Predictive Control. *WSEAS Trans. Syst. Control* **2016**, *11*, 371–375.
30. Wang, H.; Li, S.; Lan, Q.; Zhao, Z.; Zhou, X. Continuous terminal sliding mode control with extended state observer for PMSM speed regulation system. *Trans. Inst. Meas. Control* **2017**, *39*, 1195–1204. [\[CrossRef\]](#)
31. Piao, J.; Kim, E.; Choi, H.; Moon, C.; Choi, E.; Park, J.; Kim, C.-S. Indirect Force Control of a Cable-Driven Parallel Robot Tension Estimation using Artificial Neural Network trained by Force Sensor Measurements. *Sensors* **2019**, *19*, 2520. [\[CrossRef\]](#)
32. Jin, X.; Jung, J.; Ko, S.Y.; Choi, E.; Park, J.-O.; Kim, C.-S. Geometric Parameter Calibration for a Cable-Driven Parallel Robot Based on a Single One-Dimensional Laser Distance Sensor Measurement and Experimental Modeling. *Sensors* **2018**, *18*, 2392.
33. Piao, J.; Jin, X.; Jung, J.; Choi, E.; Park, J.-O.; Kim, C.-S. Open-loop position control of a polymer cable-driven parallel robot via a viscoelastic cable model for high payload workspaces. *Adv. Mech. Eng.* **2017**, *9*, 1–12. [\[CrossRef\]](#)
34. Cafolla, D.; Russo, M.; Carbone, G. CUBE, a Cable-driven Device for Limb Rehabilitation. *J. Bionic Eng.* **2019**, *16*, 493–494. [\[CrossRef\]](#)
35. Kumar, R.; Mukherjee, S. Enhanced Dynamic Capability of Cable-Driven Parallel Manipulators by Reconfiguration. *Robotica* **2021**, 1–9. [\[CrossRef\]](#)
36. Mu, Z.; Liu, T.; Xu, W.; Lou, Y.; Liang, B. Dynamic feedforward control of spatial cable-driven hyper-redundant manipulators for on-orbit servicing. *Robotica* **2019**, *37*, 18–38. [\[CrossRef\]](#)

- 
37. Yao, S.; Ceccarelli, M.; Carbone, G.; Dong, Z. Grasp configuration planning for a low-cost and easy-operation under actuated three-fingered robot hand. *Mech. Mach. Theory* **2018**, *129*, 51–69. [[CrossRef](#)]
  38. Gherman, B.; Birlescu, I.; Plitea, N.; Carbone, G.; Tarnita, D.; Pislă, D. On the singularity-free workspace of a parallel robot for lower-limb rehabilitation. *Proc. Roman. Acad. Series A Math. Phys. Techn. Sci. Inform. Sci.* **2019**, *20*, 383–391.
  39. Boschetti, G.; Carbone, G.; Passarini, C. Cable failure operation strategy for a rehabilitation cable-driven robot. *Robotics* **2019**, *8*, 17. [[CrossRef](#)]
  40. Hernández-Martínez, E.E.; Ceccarelli, M.; Carbone, G.; López-Cajún, C.S.; Jáuregui-Correa, J.C. Characterization of a cable-based parallel mechanism for measurement purposes. *Mech. Based Design Struct. Mach.* **2010**, *38*, 25–49. [[CrossRef](#)]
  41. Tucan, P.; Vaida, C.; Ulinici, I.; Banica, A.; Burz, A.; Pop, N.; Birlescu, I.; Gherman, B.; Plitea, N.; Antal, T.; et al. Optimization of the ASPIRE Spherical Parallel Rehabilitation Robot Based on Its Clinical Evaluation. *Int. J. Environ. Res. Public Health* **2021**, *18*, 3281. [[CrossRef](#)]
  42. Vaida, C.; Birlescu, I.; Pislă, A.; Ulinici, I.; Tarnita, D.; Carbone, G.; Pislă, D. Systematic design of a parallel robotic system for lower limb rehabilitation. *IEEE Access* **2020**, *8*, 34522–34537. [[CrossRef](#)]
  43. Ceccarelli, M.; Carbone, G.; Ottaviano, E. Multi criteria optimum design of manipulators. *Bull. Polish Acad. Sci. Techn. Sci.* **2019**, *53*, 9–18.