Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

A new control scheme of cable-driven parallel robot balancing between sliding mode and linear feedback

Etienne Picard * **Elias Tahoumi** ** **Franck Plestan** **
Stéphane Caro *** **Fabien Claveau** ****

* *IRT Jules Verne, Bouguenais, 44340, France (e-mail: etienne.picard@irt-jules-verne.fr).*
** *École Centrale de Nantes - LS2N, UMR CNRS 6004, Nantes, 44321 France (e-mail: franck.plestan@ec-nantes.fr).*
*** *Centre National de la Recherche Scientifique (CNRS), Nantes, 44321 France (e-mail: stephane.caro@ls2n.fr).*
**** *IMT-Atlantique - LS2N, UMR CNRS 6004, Nantes, 44300 France (e-mail: fabien.claveau@imt-atlantique.fr).*

Abstract: This paper deals with the design of a robust control scheme for a suspended Cable-Driven Parallel Robot (CDPR), composed of eight cables and a moving-platform (MP), for a pick-and-place application of metal plates of various shapes, sizes and masses. The set composed of the MP and a metal plate can have a mass of up to 700 kg. In order to achieve good accuracy and repeatability of the MP pose despite the variability of the transported mass, a robust control scheme must be implemented on the robot. A recently developed controller balancing between sliding mode and linear algorithms (SML) is considered for the application. The performances of the SML controller are analyzed on a CDPR prototype located at IRT Jules Verne, Nantes, France, along a test trajectory for several payloads. The results obtained without any information on the platform or metal plate mass are compared to those of standard proportional-derivative (PD) based control schemes.

Keywords: Cable-driven parallel robot, Robust control, Sliding mode, Experimental results

1. INTRODUCTION

Cable-Driven Parallel Robots (CDPRs) are a particular class of parallel robots whose moving-platform (MP) is connected to a fixed base frame by cables, as illustrated in Fig. 1. The cables are coiled on motorized winches. Passive pulleys may guide the cables from the winches to the cable exit points. Accordingly, the motion of the MP is controlled by modifying the cable lengths. CDPRs have several advantages such as a relatively low mass of moving parts and a potential large workspace. As a consequence, they can be used in several applications such as heavy load handling (Albus et al., 1992), painting and sand-blasting of large structures (Gagliardini et al., 2018), fast pick-and-place operations (Kawamura et al., 2000), haptic devices (Fortin-Coté et al., 2014), support structures for giant telescopes (Yao et al., 2010), and search and rescue deployable platforms (Merlet and Daney, 2010). It should be noticed that redundant actuated CDPRs are more appropriate than cranes for accurate pick-and-place operations and large and heavy parts because they suffer less from load swinging. Moreover, CDPRs can control both the position and the orientation of the object contrary to standard cranes. Accordingly, this paper deals with the determination of a control solution for a suspended semi-industrial CDPR prototype for pick and place operations of metal plates. Due to the variability of the load, robust control is required to get high accuracy and repeatability of the MP pose.

CDPR control strategies are often based on PD controllers (Kawamura et al., 2000), which can be completed with feedforward terms to predict the moving-platform dynamic behavior (Lamaury et al., 2013; Santos et al., 2019). However, the tuning of linear controllers for non-linear systems is delicate due to their hands-on tuning methods. Moreover, the computation of the feedforward term requires some levels of knowledge on the carried load, which is not always available. CDPRs, which have more actuators than degrees of freedom, are over-actuated nonlinear systems. Nonlinear control methods, including the more recent developments on sliding-mode controllers, are particularly interesting due to their robustness to uncertainties and perturbations (Edwards and Spurgeon, 1998). Sliding mode control has been increasingly considered for CDPR control in several applications (Zeinali and Khajepour, 2010; El-Ghazaly et al., 2015; Santos et al., 2019) both in simulation and experimentally, with good performances against perturbations. The drawback of sliding mode control is the existence of discontinuities in the control input due to the use of the sign function (Utkin, 1992; Shtessel et al., 2014). As a consequence, the *chattering* phenomenon appears: it is a high-frequency oscillation that leads to vibrations on the actuators and can prematurely deteriorate gearheads and other mobile parts in the kinematic chains. Higher-order and gain-adaptive sliding mode control methods have been developed to reduce chattering (Utkin, 1992; Shtessel et al., 2014; Levant, 1993), and have been implemented on a CDPR in (Schenk et al., 2018). Another drawback of sliding mode control is that the power consumption is generally higher than with linear control methods as the system is constantly excited to achieve high tracking accuracy. Recently, new control methods based on linear and sliding mode algorithms have been developed to achieve both lower chattering and energy consumption compared to pure sliding mode controllers (Tahoumi et al., 2018b). The controller then balances between the two control types to get a good trade-off between robustness and smoothness of the control output. In the sequel, this controller is defined as the sliding-mode/linear (SML) controller. The objective of the SML controller is to take advantage of both control strategies: *i)* reduced chattering and energy consumption compared to sliding mode control and *ii)* accuracy, stability and robustness despite perturbations and uncertainties.

The objective of the paper is to implement a robust control scheme suited for the considered pick-and-place application. As such, the novel controller based on sliding mode and linear algorithms (PC1-SML), is experimentally compared to a simple PD controller (PC1-PD), and to a control scheme implementing a PD controller with a feedforward term that compensates for the MP mass (PC2-PD). First, the empty MP of known mass is moved along a test trajectory. Then the trajectory is repeated while carrying metal plates of unknown mass, that constitute a perturbation to the system: a metal plate M1 of mass equal to 122 kg, then a metal plate M2 of mass equal to 249 kg.

The paper is organized as follows. Section 2 presents the CDPR semi-industrial prototype used in the experiments as well as its modeling. Section 3 describes the experimental setup and the test trajectory. The control scheme and controllers are detailed in Sec. 4. Experimental results are presented and analyzed in Section 5. Finally, conclusions are drawn and future work is presented in Section 6.

2. PROTOTYPE DESCRIPTION AND MODELING

This section deals with the description and modeling of the CDPR prototype named CAROCA (Gagliardini et al., 2018), used for the experimental comparison of the control schemes and shown in Fig. 1.

2.1 CAROCA prototype and ROCKET project

CAROCA is a reconfigurable CDPR prototype developed at IRT Jules Verne, Nantes, France, dedicated to industrial operations. A video of a logistics application on the prototype is available [1] . It this paper, its application is the displacement of metal plates of highly variable shape and mass, up to 700 kg, with a targeted accuracy of 1 cm and a desired repeatability of 1 mm.

This prototype is reconfigurable, because its pulleys can be displaced in a discrete manner on its frame, allowing the robot to be mounted both in a suspended configuration and in a fully-constrained configuration depending on the targeted application. In this paper, only the suspended configuration is considered. The size of the prototype is 7 m long, 4 m wide, and 3 m high. It is composed of 8 cables coiled around 120 mm diameter Huchez$^{\text{TM}}$ winches, that are pulling a moving-platform. The winches are actuated by B&R Automation$^{\text{TM}}$ synchronous motors of nominal speed and nominal torques equal to 2200 rpm and 15.34 Nm, respectively. A two-stage gearbox of reduction ratio equal to 40 is assembled between each motor and each winch. As a consequence, the prototype can carry up to 1 ton. Figure 1 presents the moving-platform (MP) of size 1.5 m$\times$1.5 m$\times$1 m and mass 366 kg. Five magnets are embedded under the moving platform to pick metal parts. The robot is also equipped with Tractel$^{\text{TM}}$ force sensors located between the cables and the anchor points of the platform (Fig. 1). Hardware such as motors and control bay are standard industrial components commercialized by B&R Automation$^{\text{TM}}$. The robot programming is done under Automation Studio 4.1$^{\text{TM}}$ and runs in a 2 ms real-time loop (500 Hz).

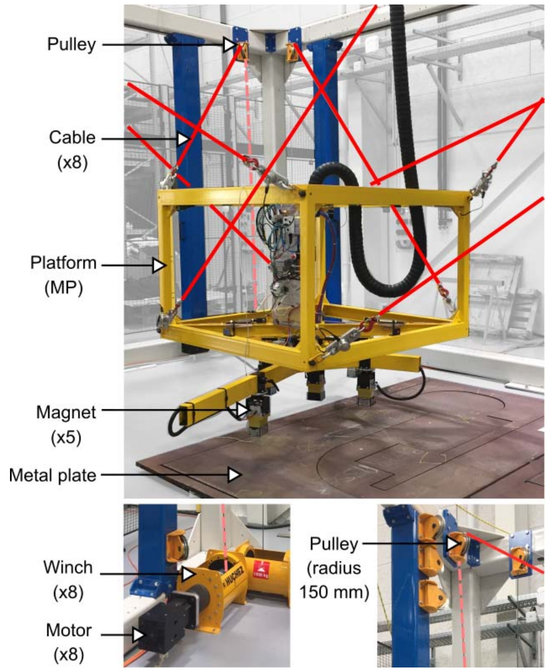


Fig. 1. The moving-platform (MP) of mass equal to 366 kg, equipped with five magnets to pick metal plates.

2.2 Inverse Geometric Model (IGM)

Figure 2 depicts the main geometric parameters of a CDPR and its i^{th} loop-closure equation, $i \in \{1, \ldots, m\}$, m being the number of cables attached to the MP, $\mathscr{F}_b$ is the robot base frame, and $\mathscr{F}_p$ is the MP frame. Cable exit points are denoted as A_i, while cable anchor points are denoted as B_i. Vector ${}^b\mathbf{a}_i$ points from O to A_i and is expressed in frame $\mathscr{F}_b$. Vector ${}^p\mathbf{b}_i$ point from P to B_i and is expressed in frame $\mathscr{F}_p$. Vector ${}^b\mathbf{p}$ is the position vector of point P, the MP geometric center, expressed in $\mathscr{F}_b$.

Vector $\mathbf{l}_i$ represents the i^{th} cable vector and points from B_i to A_i, and reads as :

[1] CDPR logistics application at IRT Jules Verne (YouTube): bit.ly/irtjvlogisticscdpr

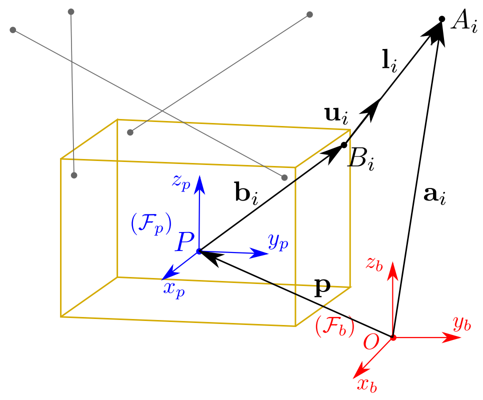


Fig. 2. CDPR geometric parameterization.

$$ {}^b\mathbf{l}_i = l_i {}^b\mathbf{u}_i = {}^b\mathbf{a}_i - {}^b\mathbf{p} - {}^b\mathbf{R}_p {}^p\mathbf{b_i} \tag{1} $$

with ${}^b\mathbf{R}_p$ the rotation matrix from frame $\mathscr{F}_b$ to frame $\mathscr{F}_p$. l_i is the length of the i^{th} cable and $\mathbf{u}_i$ is the unit vector of the ith cable vector, defined as

$$ l_i = \|{}^b\mathbf{l}_i\|_2 \qquad\qquad {}^b\mathbf{u}_i = \frac{{}^b\mathbf{l}_i}{\|{}^b\mathbf{l}_i\|_2} \tag{2} $$

where $\|.\|_2$ denotes the Euclidean norm of a vector.

To benefit from the most accurate modeling, the CDPR pulleys can be included in the geometric model of the CDPR, as described in (Gagliardini, 2016; Picard et al., 2018).

2.3 Static equilibrium

The static equilibrium of the platform is given by

$$ \mathbf{Wt} + \mathbf{w}_e + \mathbf{w}_g = 0 \tag{3} $$

with $\mathbf{W}$ the wrench matrix of the robot and expressed as

$$ \mathbf{W} = \begin{bmatrix} {}^b\mathbf{u}_1 & \dots & {}^b\mathbf{u}_i & \dots & {}^b\mathbf{u}_m \\ {}^b\mathbf{b}_1 \times {}^b\mathbf{u}_1 & \dots & {}^b\mathbf{b}_i \times {}^b\mathbf{u}_i & \dots & {}^b\mathbf{b}_m \times {}^b\mathbf{u}_m \end{bmatrix} \tag{4} $$

$\mathbf{t}$ is the cable tension vector. $\mathbf{w}_g$ the wrench applied to the platform due to gravity and $\mathbf{w}_e$ an external wrench expressed in frame $\mathscr{F}_b$.

2.4 Inverse Kinematic Model (IKM)

For CDPRs, the forward Jacobian matrix $\mathbf{A}$ relates the MP twist $\mathbf{v}$ and the cable unwinding velocities:

$$ \mathbf{Av} = \dot{\mathbf{l}} = \frac{r_w}{R}\dot{\mathbf{q}} \qquad \text{with} \qquad \mathbf{v} = \begin{bmatrix} {}^b\dot{\mathbf{p}} & {}^b\boldsymbol{\omega} \end{bmatrix}^T \tag{5} $$

$\dot{\mathbf{l}} = [\dot{l}_1 \dots \dot{l}_i \dots \dot{l}_8]^{\mathrm{T}}$ being the vector containing the cable velocities, $\dot{\mathbf{q}} = [\dot{q}_1 \dots \dot{q}_i \dots \dot{q}_8]^{\mathrm{T}}$ being the vector containing the motor velocities, R the gearbox reduction ratio, r_w the winch radius, ${}^b\mathbf{p}$ the Cartesian position of MP and ${}^b\boldsymbol{\omega}$ its angular velocity, expressed in $\mathscr{F}_b$. $\mathbf{A}$ and $\mathbf{W}$ are related by the equation:

$$ \mathbf{W} = -\mathbf{A}^T \tag{6} $$

2.5 Dynamic model

From (Gagliardini et al., 2018), the dynamic model of the CDPR reads as

$$ \mathbf{Wt} - \mathbb{I}_p\dot{\mathbf{v}} - \mathbf{Cv} + \mathbf{w}_e + \mathbf{w}_g = 0 \tag{7} $$

with $\mathbb{I}_p$ the spatial inertia of the platform and $\mathbf{C}$ the matrix of the centrifugal and Coriolis wrenches.

Given that the center of mass of the platform G does not coincide with the origin of $\mathscr{F}_p$, the wrench $\mathbf{w}_g$ due to the gravity acceleration $\mathbf{g}$ is defined as

$$ \mathbf{w}_g = \begin{bmatrix} m_p\mathbf{I}_3 \\ \mathbf{M}\hat{\mathbf{S}}_p \end{bmatrix} \mathbf{g} \tag{8} $$

with m_p the mass of the platform, $\mathbf{I}_3$ the 3×3 identity matrix, $\mathbf{MS}_p ={}^b \mathbf{R}_p \, [m_p x_G \; m_p y_G \; m_p z_G]^{\mathrm{T}}$ the first momentum of the moving platform defined with respect to frame $\mathscr{F}_b$. The vector $\mathbf{S}_p = [x_G \; y_G \; z_G]^{\mathrm{T}}$ defines the position of G in $\mathscr{F}_p$. $\mathbf{M}\hat{\mathbf{S}}_p$ is the skew-symmetric matrix associated to $\mathbf{MS}_p$.

$\mathbb{I}_p$ represents the spatial inertia of the platform, and reads as

$$ \mathbb{I}_p = \begin{bmatrix} m_p\mathbf{I}_3 & -\mathbf{M}\hat{\mathbf{S}}_p \\ \mathbf{M}\hat{\mathbf{S}}_p & \mathbf{I}_p \end{bmatrix} \tag{9} $$

with $\mathbf{I}_p$ the inertia tensor matrix of the platform, that can be computed from the platform's inertia tensor $\mathbf{I}_g$ using the Huygens-Steiner theorem

$$ \mathbf{I}_p ={}^b \mathbf{R}_p \, \mathbf{I}_g \, {}^b\mathbf{R}_p^{\mathrm{T}} - \frac{\mathbf{M}\hat{\mathbf{S}}_p\mathbf{M}\hat{\mathbf{S}}_p}{m_p} \tag{10} $$

$\mathbf{C}$ is the matrix of the centrifugal and Coriolis wrenches with

$$ \mathbf{Cv} = \begin{bmatrix} {}^b\hat{\boldsymbol{\omega}}{}^b\hat{\boldsymbol{\omega}}\mathbf{MS}_p \\ {}^b\hat{\boldsymbol{\omega}}\mathbf{I}_p{}^b\boldsymbol{\omega} \end{bmatrix} \tag{11} $$

where ${}^b\hat{\boldsymbol{\omega}}$ the skew-symmetric matrix associated to ${}^b\boldsymbol{\omega}$.

2.6 State system

The dynamics of the motors (Lamaury and Gouttefarde, 2013) are given by

$$ \boldsymbol{\tau}_m = \mathbf{I}_q\ddot{\mathbf{q}} + \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{F}_s\text{sign}(\dot{\mathbf{q}}) + \frac{R}{r_w}\mathbf{t} \tag{12} $$

where $\mathbf{I}_q$ is the diagonal matrix containing the moment of inertia of the gearmotors and winches associated to each motor, and $\mathbf{F}_c$ and $\mathbf{F}_v$ are respectively the diagonal matrices containing the static and viscous friction coefficients for each motor.

From the motor dynamic model (equation 12), the CDPR inverse kinematic model (equation (5)) and the CDPR dynamic model (equation (7)), defining the state vector as $\mathbf{x} = [\mathbf{q} \; \dot{\mathbf{q}}]^{\mathrm{T}}$ and the system input as $\mathbf{u} = \boldsymbol{\tau}_m$, the system can be represented as a standard nonlinear system of the form

$$ \mathbf{x} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \tag{13} $$

The system is nonlinear and affine in the control input $\mathbf{u}$. Furthermore, $f(\mathbf{x})$ is uncertain due to the presence of $\mathbf{w}_e$ in (equation (7)).

3. TEST TRAJECTORY AND EXPERIMENTAL SETUP

In order to evaluate the performance of different control methods, a desired trajectory describing a typical pick-and-place application has been generated. The trajectory is generated using s-curves, that ensure continuous velocity and acceleration trajectory profiles. The x-axis of the frame $\mathscr{F}_b$ is defined along the width of the CDPR, the

y-axis along its length, and the z-axis along its height. The trajectory consists of (see Fig. 3):

(1) AB: 200 mm vertical displacement up;
(2) BC: arc along the diagonal of the base footprint, with simultaneous displacements of 300 mm up, 300 mm along the x-axis and 1400 mm along the y-axis;
(3) CD: arc along the diagonal of the base footprint, with simultaneous displacements of 300 mm down, 300 mm along the x-axis and 1400 mm along the y-axis;
(4) DE: 200 mm vertical displacement down;

The platform moves from A to E in 30 s.

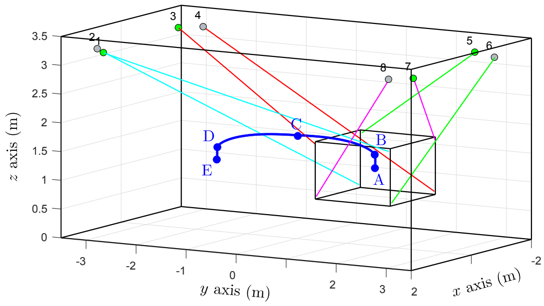


Fig. 3. Test trajectory (blue) and CDPR configuration.

This test is first performed on the CDPR with the empty platform of mass 366 kg. Then, to evaluate the control robustness, two metal plates are successively carried by the platform. Three cases are then considered:

(1) the empty moving-platform of mass 366 kg (MP);
(2) the MP and a metal plate M1 of mass 122 kg, for a total load of 488 kg (MPM1);
(3) the MP and a metal plate M2 of mass 249 kg, for a total load of 615 kg (MPM2)

Note that the mass variation is significant, namely +33% (M1) and +68% (M2) with respect to the MP mass, respectively.

(a) MPM1 (488 kg). (b) MPM2 (615 kg).

Fig. 4. MP carrying a metal plate: (a) M1, (b) M2.

4. CONTROL STRATEGIES

Accordingly to equation (12), Fig. 5 presents the system input, output and perturbation signals. The system input is the motor torque vector $\boldsymbol{\tau}_m$. The usual outputs of the system are the actual motor positions $\mathbf{q}$ and motor velocities $\dot{\mathbf{q}}$. The considered perturbation is gravity wrench $\mathbf{w}_g$ of the moving-platform and the embedded metal plate.

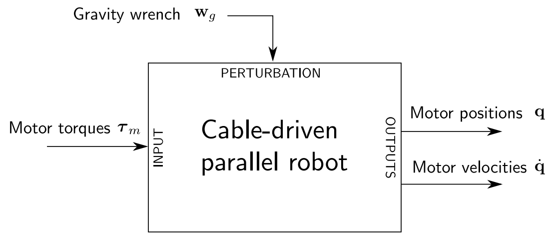


Fig. 5. Diagram of the CDPR.

4.1 Control schemes

No direct information on the platform pose is readily available from the system sensors, and solving the direct geometric model of a CDPR is not an easy task since more than one solution is possible from a fixed set of motor positions, even considering straight and inelastic cables (Merlet, 2015). As a consequence, the following control architectures only rely on the system internal sensors *i.e.* the motor angular positions and velocities. Decentralized control architectures have been considered for their simplicity of implementation, with one controller separately tuned for each motor.

PC1: Basic control scheme. The first control architecture is denoted as PC1 (Fig. 6). The *controller* box at the center of the control schemes is left unspecified in this section. In the sequel, the proportional-derivative controller and the sliding mode based controller will be introduced. The corresponding control architecture is then referenced as PC1-PD or PC1-SML, accordingly.

$\mathbf{c}_d$ is the 6-dimensional vector containing the desired Cartesian position and orientation of the MP, $\mathbf{v}_d$ the desired MP twist (linear and angular platform velocities). $\mathbf{q}_d$, $\dot{\mathbf{q}}_d$ and $\boldsymbol{\tau}_m$ are the desired motor angular positions, velocities obtained from the inverse geometric (IGM) and kinematic (IKM) models, respectively. In this first scheme, the motor torque vector $\boldsymbol{\tau}_m$ is the control signal, each signal being of dimension 8.

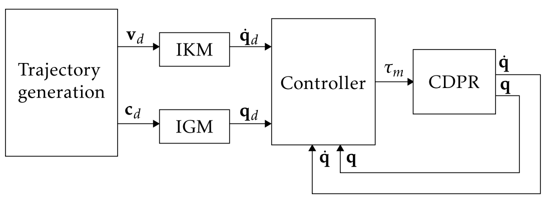


Fig. 6. PC1 control architecture.

The MP Cartesian MP pose and twist are converted into desired motor positions and velocities using the CDPR inverse geometric (IGM) and kinematic (IKM) models.

PC2: Control scheme with feedforward. Feedforward terms (see Fig. 7 blue blocks) are commonly included in CDPR control strategies to predict the dynamics of the platform and improves the accuracy of the robot (Lamaury et al., 2013; Vafaei et al., 2010).

From Eq. (7), a feedforward term compensating part of the gravity is defined as

$$\boldsymbol{\tau}_{da} = \frac{r_w \mathbf{W}^{\dagger}(\mathbb{I}_p \dot{\mathbf{v}}_d + \mathbf{w}_g)}{R} \qquad (14)$$

with τ_{da} the feedforward torque, $\mathbf{w}_g$ the wrench due to the gravity, r_w the radius of the winches and R the gearhead ratio. $\mathbf{W}^\dagger$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{W}$. However, for metal plate handling,the mass of the metal plates is supposed to be unknown. As a consequence, only the MP mass is considered in the feedforward term, that gives

$$\mathbf{w}_g = [0\ 0\ -m_{MP}g\ -m_{MP}gy_G\ m_{MP}gx_G\ 0]^{\mathrm{T}} \tag{15}$$

with m_{MP} the MP mass and $\mathbf{g}$ the gravity vector expressed in $\mathscr{F}_b$. x_G and y_G are the Cartesian coordinates of the MP center of gravity G expressed in $\mathscr{F}_p$. Note that x_G and y_G are supposed to be null along the trajectory.

Also, a linear friction model (Khalil and Dombre, 2004) has been implemented in each actuation chain to compensate the losses in the motors, gearbox and winches:

$$\boldsymbol{\tau}_{fc} = \mathbf{F}_c\mathrm{sign}(\dot{\mathbf{q}}_d) + \mathbf{F}_v\dot{\mathbf{q}}_d \tag{16}$$

with $\boldsymbol{\tau}_{fc}$ the friction compensation and $\dot{\mathbf{q}}_d$ the desired motor rate vector.

Figure 7 presents the PC2 control architecture with feedforward, where $\dot{\mathbf{v}}_d$ contains the Cartesian acceleration and angular acceleration of the platform. The control torques in $\boldsymbol{\tau}_m$ applied to the motors is based finally on $\boldsymbol{\tau}_c$, $\boldsymbol{\tau}_{da}$ and $\boldsymbol{\tau}_{fc}$.

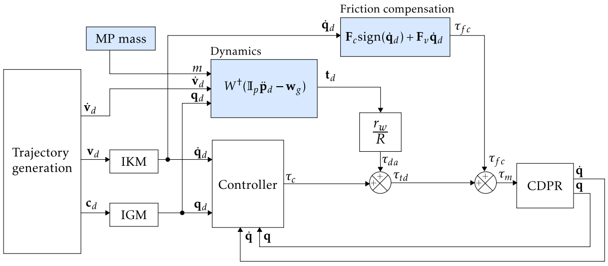


Fig. 7. PC2 control architecture with feedforward terms.

4.2 Control algorithms

The control algorithms are described thereafter.

PD controller. The 8-dimensional output signal of the controller $\boldsymbol{\tau}_{\mathrm{c}}$ is therefore defined by

$$\boldsymbol{\tau}_{\mathrm{c}} = \mathbf{K}_{\mathrm{p}}\mathbf{e}_q + \mathbf{K}_{\mathrm{d}}\mathbf{e}_{\dot{q}} \tag{17}$$

with $\mathbf{e}_q$ the difference between the desired and actual motor positions and $e_{\dot{\mathbf{q}}}$ being the difference between the desired and actual motors velocities. In PC1-PD (Fig. 6), $\mathbf{u} = \boldsymbol{\tau}_{\mathrm{m}} = \boldsymbol{\tau}_{\mathrm{c}}$, while in PC2-PD (Fig. 7), $\mathbf{u} = \boldsymbol{\tau}_{\mathrm{m}} = \boldsymbol{\tau}_{\mathrm{c}} + \boldsymbol{\tau}_{\mathrm{da}} + \boldsymbol{\tau}_{\mathrm{fc}}$.

In a decentralized control architecture, each motor is independently controlled: then, the matrices $\mathbf{K}_{\mathrm{p}}$ and $\mathbf{K}_{\mathrm{d}}$ are diagonal. For simplicity and since the identified motor friction coefficients are similar across all motors, the 8 decentralized controllers have been tuned similarly: $K_p = K_{p,1} = K_{p,2}\ldots$ and $K_d = K_{d,1} = K_{d,2}\ldots$. However, it could be possible to independently adjust the gains of each motor according to their errors along the test trajectory or interdependence. The PD controller has been tuned to achieve accuracy and stability with the MP, using the standard method proposed by (Ziegler and Nichols, 1995). The obtained gains K_p and K_d are given in Table 1.

Table 1. PD controller gains.

Gain	K_p	K_d
ROMP values	0.3	0.03

SML controller. Similarly to the PD controller, eight individual SML controllers have been implemented. Note that in this paper, the SML controller is applied only to the PC1 scheme (Fig. 6).

Define the sliding vector $\boldsymbol{\sigma}$ as

$$\boldsymbol{\sigma} = (\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \lambda(\mathbf{q}_d - \mathbf{q}) \tag{18}$$
$$= \mathbf{e}_{\dot{q}} + \lambda\mathbf{e}_q \tag{19}$$

with $\mathbf{q}_d$ and $\mathbf{q}$ respectively the desired and current motor angular positions, $\dot{\mathbf{q}}_d$ and $\dot{\mathbf{q}}$ respectively the desired and current motor velocities, $\mathbf{e}_q$ and $\mathbf{e}_{\dot{q}}$ the corresponding tracking errors and λ a strictly positive parameter ($\lambda > 0$).

Sliding mode control must ensure that the sliding variable reaches and is maintained at zero in a finite time (Utkin, 1992; Shtessel et al., 2014): given the definition (18) of $\boldsymbol{\sigma}$, when the sliding variable of the i^{th} motor σ_i tends to zero, the convergence of $e_{q,i}$, the i^{th} component of $\mathbf{e}_q$, to zero is guaranteed exponentially with a rate depending on the parameter λ. This is described as the *transient phase*. Then, the controller is in the steady state: the sliding variable σ_i is maintained around zero and the dynamic of the control is defined by the differential equation $e_{\dot{q},i} = -\lambda e_{q,i}$, as such the higher λ, the faster the correction. This is described as the *sliding phase*.

$\boldsymbol{\sigma}$ has a relative degree of one with respect to $\boldsymbol{\tau}_m$. The time derivative of the sliding variable equals

$$\dot{\boldsymbol{\sigma}} = \mathbf{e}_{\ddot{q}} + \lambda\mathbf{e}_{\dot{q}} \tag{20}$$
$$= (\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}) + \lambda(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \tag{21}$$
$$= (\ddot{\mathbf{q}}_d + \lambda(\dot{\mathbf{q}}_d - \dot{\mathbf{q}})) - \ddot{\mathbf{q}} \tag{22}$$

$\ddot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}$ being the desired and actual motor acceleration vectors, respectively. $\ddot{\mathbf{q}}$ is correlated to the motor torques $\boldsymbol{\tau}_m$ by equation (12). $\dot{\boldsymbol{\sigma}}$ then takes the form

$$\dot{\boldsymbol{\sigma}} = a(\mathbf{q}) + b(\mathbf{q})\boldsymbol{\tau}_m \tag{23}$$

where $b(\mathbf{q}) \neq 0$.

In order to design the twisting algorithm, the sliding variable is derived a second time ; one gets:

$$\ddot{\boldsymbol{\sigma}} = (\dddot{\mathbf{q}}_d - \dddot{\mathbf{q}}) + \lambda(\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}) \tag{24}$$
$$= (\dddot{\mathbf{q}}_d + \lambda(\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}})) - \dddot{\mathbf{q}} \tag{25}$$
$$= h(\mathbf{q}) + j(\mathbf{q})\dot{\boldsymbol{\tau}}_m \tag{26}$$

with $j(\mathbf{q}) \neq 0$.

Each component of $\dot{\boldsymbol{\tau}}_m$ Tahoumi et al. (2018a), is defined as

$$\dot{\tau}_{m,i} = -K_1\lceil\sigma_i\rfloor^{\frac{\alpha}{2-\alpha}} - K_2\lceil\dot{\sigma}_i\rfloor^{\alpha} \tag{27}$$

with $\dot{\tau}_{m,i}$ the i^{th} component of $\dot{\boldsymbol{\tau}}_m$ and

$$\lceil\sigma_i\rfloor^{\alpha} = |\sigma_i|^{\alpha}\mathrm{sign}(\sigma_i) \tag{28}$$

K_1 and K_2 are the controller gains, and $\alpha \in [0\ 1]$ based on the following adaptation law:

$$\alpha = \max\left(-\beta\left(\frac{|\sigma_i|}{|\sigma_i| + \epsilon_\sigma} + \frac{|\dot{\sigma}_i|}{|\dot{\sigma}_i| + \epsilon_{\dot{\sigma}}}\right) + 1, 0\right) \tag{29}$$

with β, ϵ_σ and $\epsilon_{\dot{\sigma}}$ constant parameters chosen such that $\beta > 1$ and $\epsilon_\sigma, \epsilon_{\dot{\sigma}} > 0$. Values of these parameters for the experiments are provided in table 2. The control input $\boldsymbol{\tau}_m$

is then obtained by integrating its time derivative from Eq. (27).

The principle of the SML controller (Eqs. (27)-(29)) is the following: the value of the variable α depends on the current tracking errors. If the absolute values of $|\sigma_i|$ and $|\dot{\sigma}_i|$ are large, it means that the closed-loop system is not accurate: the controller should lean towards a robust controller, namely the sliding mode control. That is the case because in such a situation, $\alpha \to 0$, from Eq. (27), the control becomes a twisting one (Levant, 1993):

$$\dot{\tau}_{m,i} = -K_1 \text{sign}(\sigma_i) - K_2 \text{sign}(\dot{\sigma}_i) \tag{30}$$

that ensures, in practice, the convergence of σ_i and $\dot{\sigma}_i$ to a vicinity of $(0, 0)$, in finite time.

On the other hand, if these errors are small, in order to reduce chattering and energy consumption, the controller should lean towards the linear control behavior: that is the case because $\alpha \to 1$. α regulates the trade-off between accuracy and chattering reduction. $\dot{\tau}_{m,i}$ then tends towards the expression

$$\dot{\tau}_{m,i} = -K_1 \sigma_i - K_2 \dot{\sigma}_i \tag{31}$$

In order to guarantee convergence of the closed-loop system, the gains K_1 and K_2 must be positive and follow the condition (Levant, 1993):

$$\begin{aligned} K_1 > K_2 > 0, \quad & (K_1 - K_2) j_m > h_M \\ & (K_1 + K_2) j_m - h_M > (K_1 - K_2) j_m + h_M \end{aligned} \tag{32}$$

with h_M, j_m and j_M positive constants such that for each motor

$$|h_i(q)| \leqslant h_M \tag{33}$$

$$0 < j_m \leqslant j_i(q) \leqslant j_M \tag{34}$$

The reduced energy consumption of the SML controller compared to the twisting algorithm is ensured by the following condition (Tahoumi et al., 2018a):

$$K_1 \epsilon_\sigma + K_2 \epsilon_{\dot{\sigma}} < K_1 - K_2 \tag{35}$$

Recall that K_1 and K_2 are the gains of the controller: they must be chosen sufficiently large to counteract perturbations and uncertainties effects. β and ϵ_σ,$\epsilon_{\dot{\sigma}}$ have opposing effects on the evolution of α. These parameters should be chosen to calibrate the controller behavior with respect to the desired compromise between accuracy and chattering/consumption reduction: the higher β or the smaller ϵ_σ, $\epsilon_{\dot{\sigma}}$, the lower α. Then, the system leans towards sliding mode. As a consequence, the control accuracy is improved with higher energy consumption. On the other hand, if β is decreased or ϵ_σ, $\epsilon_{\dot{\sigma}}$ are increased, α will increase: the chattering and energy consumption will be reduced as the linear control contribution increases. However, the robustness, and then the accuracy, are reduced.

A first tuning has been obtained from a Simulink® model based on the inverse geometric, kinematic and dynamic models of the robot. The initial working parameters are set according to the following methodology:

(1) select a value of $\lambda > 0$ in relation to the desired closed-loop dynamics and $\beta > 1$ with respect to the desired closed-loop accuracy;
(2) set ϵ_σ to zero to force α to zero and achieve pure sliding mode control;
(3) tune gains K_1 and K_2 to achieve good sliding mode control accuracy;
(4) observe $\boldsymbol{\sigma}$ and $\dot{\boldsymbol{\sigma}}$ values, choose values for ϵ_σ and $\epsilon_{\dot{\sigma}}$ with the same proportionality as the average ratio between $\boldsymbol{\sigma}$ and $\dot{\boldsymbol{\sigma}}$;
(5) adjust β, ϵ_σ and $\epsilon_{\dot{\sigma}}$ so that the variable α evolves between $[0, 1]$;

After some iterations, the gains and parameters are set to obtain a good compromise between accuracy and evolution of α. The tuning parameters for the SML controller are given in Table 2.

Table 2. SML controller parameter values.

Parameter	λ	β	ϵ_σ	$\epsilon_{\dot{\sigma}}$	K_1	K_2
ROMP values	0.15	1.01	4	80	4	2

Since only the motor position and velocity are provided on the prototype, the motor angular acceleration errors $e_{\ddot{\mathbf{q}}}$ appearing in $\dot{\boldsymbol{\sigma}}$ are derived from the motor velocity errors with the usual Euler method.

5. EXPERIMENTAL RESULTS

The objective of the experiments is to compare the performances of the SML controller without knowledge of the MP and metal plate mass (PC1-SML), to those of the PD controller (PC1-PD) in the same conditions, and of the PD controller with feedforward for the compensation of the MP mass (PC2-PD). A video of the metal plate handling experiments is available [2].

5.1 Motor position errors

Figure 8 presents the *Root Mean Square* (RMS) of Motors 1 to 8 position errors along the test trajectory, for each controller and load. As expected, the PC1-PD gives the highest RMS in all scenarios, up to 12 degrees, due to the controller static error. PC2-PD provides the smallest RMS for load MP, as the mass of the platform is exactly compensated by the feedforward term. Thus, the static error is greatly reduced. However, the RMS of PC2-PD motor position error naturally increases as a metal plate of unknown mass is carried. It appears that PC1-SML provides the most consistent results no matter the load, the motor position errors being always smaller than 4 degrees. Slight differences in the performances from one motor to another could be attributed to the impact of the new center of mass of the set constituted of the MP and the metal plate.

A single motor is considered in the sequel for a more detailed analysis. Fig. 9 presents Motor 4 angular error e_{q4} along the trajectory, with the three controllers and the three payloads. Again, the static errors of PC1-PD and PC2-PD are visible as the mass increases. The maximum error is reached around 11 degrees for the heaviest load (MPM2) for PC1-PD. The compensation of the MP mass limits this maximum error to around 4 degrees for PC2-PD. Although $e_{q,4}$ has a more oscillating behavior with PC1-SML, it is noteworthy that it is the most robust one amongst the three controllers. Indeed, its value oscillates around zero for all loads.

[2] Metal plate handling video (Dropbox): bit.ly/ifac2020id3263

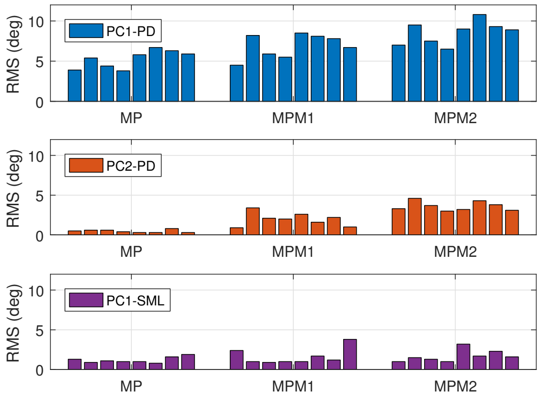
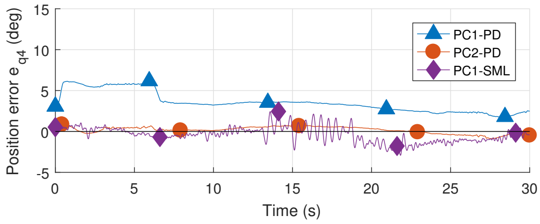
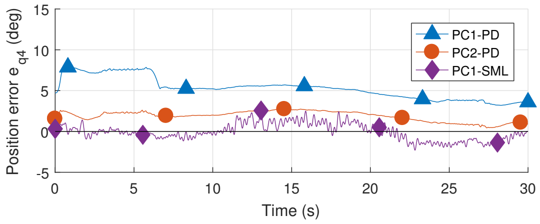


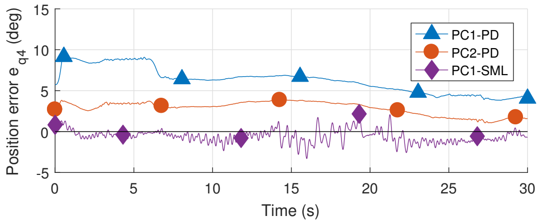
Fig. 8. *Root Mean Square* (RMS) of position error (degrees) for Motors 1 to 8 (left to right) against load.



(a) Load MP (366 kg).



(b) Load MPM1 (488 kg).



(c) Load MPM2 (615 kg).

Fig. 9. Motor 4 position error $e_{q,4}$ (degrees).

The control input of Motor 4 associated with each controller is plotted in Fig. 10. All controllers generate a similarly shaped torque output, although chattering is visible in the case of the sliding mode, that is to be expected: this is the cost of robustness. Static friction in the gearbox can be observed around $t = 1$ s and $t = 6$ s, when the motor changes direction. The friction is anticipated in the feedforward term of PC2-PD, and is quickly corrected by the SML controller, leading to similarly shaped signals.

5.2 MP position error along z-axis

The MP pose was tracked using a HTC VIVE Tracker, along the vertical axis (z-axis) which is the most affected one with the change of load. Figure 11 presents the Cartesian error between the desired and measured position of the MP center P (see Fig. 2), for the heaviest load MPM2. The controllers are compared based on the motor position errors. The PC1-PD controller leads to the largest error while the addition of the feedforward term in PC2-PD reduces this error. The sliding mode controller provides the smallest error although a static error of around 20 mm remains. This error is mainly due to cable elasticity, which is not negligible for the 615 kg load, and can be compensated in the future in order to achieve the desired accuracy of one centimeter.

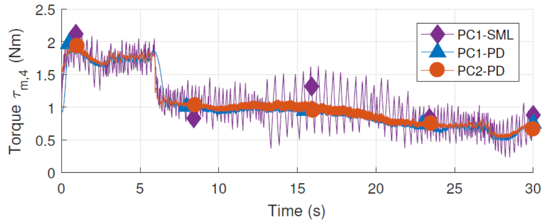


Fig. 10. Motor 4 control input τ_m (Nm) for load MPM2 (615 kg).

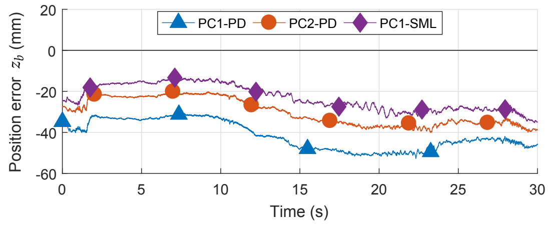


Fig. 11. MP Cartesian position error along z-axis (mm) for load MPM2 (615 kg).

5.3 Evolution of α

The mean value of α along the trajectory across all motors and for each load is presented in Fig. 12. As expected, the larger the payload, the closer the controller to sliding mode control *i.e.* the lower α. Figure 13 shows the evolution of α for Motor 4 for the heaviest load (MPM2), versus time. It can be noticed that with the current tuning of the SML controller, the values of α are relatively low with an average value of 0.12 in general, namely the controller is mostly a sliding mode one along the trajectory, due to large uncertainties in the system. It is noteworthy that around $t = 7$ s and $t = 27$ s, α reaches higher values for short periods. Meanwhile, Fig. 10 shows reduced chattering in τ_m and Fig. 9 presents lower motor position oscillations. Contrarily, around $t = 15$ s, as α lower values lead to higher oscillations. The control is effectively smoother when α increases.

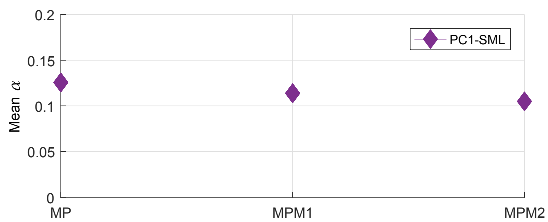


Fig. 12. Mean of α along trajectory, across all motors and for each load.

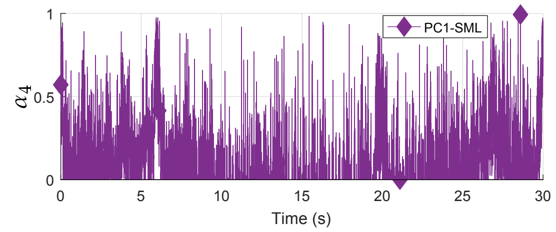


Fig. 13. Evolution of α_4 for load MPM2 (615 kg).

6. CONCLUSION

Although a proportional-derivative based control scheme can be applied for smooth control signal of a cable-driven parallel robot, it is very restrictive due to the required knowledge of the carried mass to achieve good accuracy. If no information is available on the load mass, the novel control scheme balancing between sliding mode and linear algorithms is relevant for its stability and robustness towards uncertainties, with lower chattering and oscillations as parameter α increases. Future work will focus on oscillation reduction, and include a cable elasticity compensation to improve the repeatability and positioning accuracy of the moving-platform.

ACKNOWLEDGEMENTS

The authors wish to associate the industrial and academic partners of IRT Jules Verne in the framework of the ROCKET project, namely Chantiers de l'Atlantique, Clemessy, B&R Automation, Clarté, and CNRS.

REFERENCES

Albus, J., Bostelman, R., and Dagalakis, N. (1992). The NIST SPIDER, A robot crane. *Journal of Research of the National Institute of Standards and Technology*, volume 97(3), 373.

Edwards, C. and Spurgeon, S. (1998). *Sliding mode control: theory and applications.* CRC Press.

El-Ghazaly, G., Gouttefarde, M., and Creuze, V. (2015). Adaptive terminal sliding mode control of a redundantly-actuated cable-driven parallel manipulator: CoGiRo. In *Mechanisms and Machine Science*, volume 32, 179–200. Springer, Cham.

Fortin-Coté, A., Cardou, P., and Gosselin, C. (2014). An admittance control scheme for haptic interfaces based on cable-driven parallel mechanisms. In *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, 819–925. Hong Kong.

Gagliardini, L. (2016). *Discrete reconfigurations of cable-driven parallel robots.* PhD Thesis, Centrale Nantes.

Gagliardini, L., Gouttefarde, M., and Caro, S. (2018). Determination of a Dynamic Feasible Workspace for Cable-Driven Parallel Robots. *Advances in Robot Kinematics 2016*, volume 4, 361–370.

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

Khalil, W. and Dombre, E. (2004). *Modeling, Identification and Control of Robots.* Butterworth-Heinemann.

Lamaury, J. and Gouttefarde, M. (2013). Control of a large redundantly actuated cable-suspended parallel robot. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 4659–4664. Karlsruhe, Germany.

Lamaury, J., Gouttefarde, M., Chemori, A., and Herve, P.E. (2013). Dual-space adaptive control of redundantly actuated cable-driven parallel robots. In *Proceedings of the 2013 IEEE International Conference on Intelligent Robots and Systems (IROS)*, 4879–4886. Tokyo, Japan.

Levant, A. (1993). Sliding order and sliding accuracy in sliding mode control. *International Journal of Control*, 58(6), 1247–1263.

Merlet, J.P. and Daney, D. (2010). A portable, modular parallel wire crane for rescue operations. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2834–2839.

Merlet, J.P. (2015). The kinematics of cable-driven parallel robots with sagging cables: Preliminary results. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 1593–1598. Seattle, WA, USA.

Picard, É., Caro, S., Claveau, F., and Plestan, F. (2018). Pulleys and Force Sensors Influence on Payload Estimation of Cable-Driven Parallel Robots. In *Proceedings of the 2018 IEEE International Conference on Intelligent Robots and Systems (IROS)*, 1429–1436. Madrid, Spain.

Santos, J.C., Chemori, A., and Gouttefarde, M. (2019). Model Predictive Control of Large-Dimension Cable-Driven Parallel Robots. 221–232. Springer, Cham.

Schenk, C., Masone, C., Pott, A., and Bülthoff, H.H. (2018). Application of a differentiator-based adaptive super-twisting controller for a redundant cable-driven parallel robot. In *Mechanisms and Machine Science*, volume 53, 254–267.

Shtessel, Y., Edwards, C., Fridman, L., and Levant, A. (2014). *Sliding mode control and observation.* Control Engineering. Springer New York, New York, NY.

Tahoumi, E., Ghanes, M., Plestan, F., and Barbot, J.P. (2018a). A New Controller Switching between Linear and Twisting Algorithms. *Proceedings of the 2018 American Control Conference*, 6150–6155.

Tahoumi, E., Plestan, F., Ghanes, M., and Barbot, J.P. (2018b). A Controller Switching between Twisting and Linear Algorithms for an Electropneumatic Actuator. In *Proceedings of the 2018 European Control Conference (ECC)*, 2368–2373.

Utkin, V.I. (1992). *Sliding Modes in Control and Optimization.* Springer, Berlin, Heidelberg.

Vafaei, A., Aref, M.M., and Taghirad, H.D. (2010). Integrated controller for an over-constrained cable driven parallel manipulator: KNTU CDRPM. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, 650–655.

Yao, R., Tang, X., Wang, J., and Huang, P. (2010). Dimensional optimization design of the four-cable-driven parallel manipulator in fast. *IEEE/ASME Transactions on Mechatronics*, 15(6), 932–941.

Zeinali, M. and Khajepour, A. (2010). Design and application of chattering-free sliding mode controller to cable-driven parallel robot manipulator: Theory and experiment. In *Proceedings of the 2010 ASME Design Engineering Technical Conference*, 319–327.

Ziegler, J.G. and Nichols, N.B. (1995). Optimum settings for automatic controllers. *InTech*, volume 42(6), 94–100.