



HAL
open science

Control Strategies for a Cable-Driven Parallel Robot with Varying Payload Information

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

► **To cite this version:**

Etienne Picard, Franck Plestan, Elias Tahoumi, Fabien Claveau, Stéphane Caro. Control Strategies for a Cable-Driven Parallel Robot with Varying Payload Information. *Mechatronics*, 2021, 79, pp.102648. 10.1016/j.mechatronics.2021.102648 . hal-03332227

HAL Id: hal-03332227

<https://hal.science/hal-03332227>

Submitted on 2 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Control Strategies for a Cable-Driven Parallel Robot with Varying Payload Information¹

Etienne PICARD^a, Franck PLESTAN^b, Elias TAHOUMI^b, Fabien CLAVEAU^c, Stéphane CARO^d

^aIRT Jules Verne, Bouguenais, 44340, France (e-mail: etienne.picard@irt-jules-verne.fr).

^bÉcole Centrale de Nantes - LS2N, UMR CNRS 6004, Nantes, 44321 France (e-mail: franck.plestan@ec-nantes.fr).

^cIMT-Atlantique - LS2N, UMR CNRS 6004, Nantes, 44300 France (e-mail: fabien.claveau@imt-atlantique.fr).

^dCentre National de la Recherche Scientifique (CNRS-LS2N), Nantes, 44321 France (e-mail: stephane.caro@ls2n.fr).

Abstract

A suspended Cable-Driven Parallel Robot (CDPR) composed of eight cables and a moving platform (MP) is used in a pick-and-place application of metal plates with different shapes, sizes and masses. In order to ensure robust control despite mass variation, several combinations of control schemes and control laws have been experimented on a prototype at IRT Jules Verne, France. The main objective of this paper is to provide recommendations on the selection of a control strategy depending on the available information on the carried mass, and the presence or absence of force sensors. Three scenarios are considered representing a degradation of the information on the carried mass to observe the impact on the performance of applicable control strategies. In a first case, force sensors are assumed available to measure cable tension, allowing the real-time estimation of the carried mass. In a second case, the mass of the MP is known, but not the mass of the carried metal plate whereas the third case considers no information at all on both the MP and the carried metal plate. The tested control laws include a standard proportional-derivative controller (PD), and a recently developed nonlinear controller balancing between sliding mode and linear algorithms (SML). The performances of each control strategy are analyzed along a test trajectory for several payloads, and a decision tree is proposed.

Keywords: Robust control, Cable Driven Parallel Robots, Pick-and-place operations, Handling

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 Figure 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 rela-

tively 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 [1], painting and sandblasting of large structures [7], fast pick-and-place operations [9], haptic devices [5], support structures for giant telescopes [26], and search and rescue deployable platforms [16]. It should be noticed that redundant actuated CDPRs are more appropriate than cranes for accurate pick-and-place operations of large and heavy parts

because they suffer less from load swinging. Moreover, CDPRs can control both the position and the orientation of the object.

A suspended semi-industrial CDPR prototype, named CAROCA, is used to perform pick-and-place operations of metal plates. However, due to the variability of the load, the implementation of robust control strategies is required in order to get high accuracy and repeatability. In particular, the control should be insensitive to the variability of the carried mass.

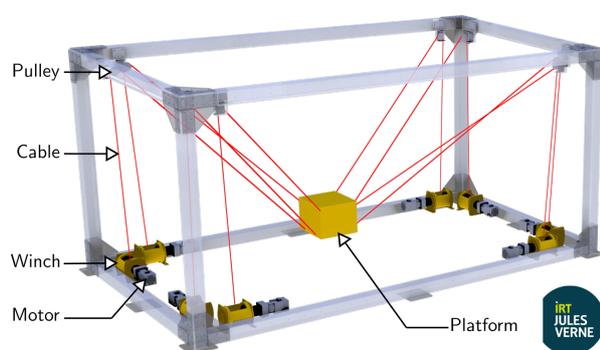


Figure 1: CAROCA prototype 3D model in a suspended configuration.

Proportional-derivative (PD) controllers are already widely used in industry and machinery. Industrial motors are usually equipped with position sensors making the implementation of a PD controller particularly simple. [The choice of a PD controller compared to the full PID was made to avoid any problem with the integral action, for stability robustness mainly, but also to avoid integrator windup \[10\].](#) Its main drawbacks are the static error between the desired and real pose of the moving-platform as well as the tuning process that often requires iteration. In addition, this controller can be sensitive to noise and its settings are optimal only around an operating point. As a consequence, if parameters such as the carried mass change, the PD controller performance might be degraded.

In the frame of CDPR control, a feedforward term can be added to predict the MP

dynamic behavior [14, 25, 19]. However, the computation of the feedforward term requires the knowledge of the carried mass. This approach will be directly applicable if the theoretical mass is perfectly known, but that might not always be the case. Sometimes only the mass of the MP could be available, or there could be no information at all on either the MP or the metal plate mass.

Some CDPRs are equipped with force sensors to measure cable tensions which can be leveraged for the estimation [12, 17] of the total mass and center of mass of the moving parts, namely the MP and the metal plate. This information can then be used in real-time for the update of the feedforward term.

CDPR are robotics systems that can be accurately modeled as nonlinear systems. In this case, nonlinear control strategies are interesting. Nonlinear control methods include the most recent developments of sliding-mode controllers, particularly interesting due to their robustness to uncertainties and perturbations [3]. Sliding mode control has been increasingly considered for CDPR control in several applications [27, 4, 19] both in simulation and experimentally. The drawback of sliding mode control is the existence of discontinuities in the control input due to the use of the sign function [24], [21]. 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 components in the actuators of the robot. Higher order and gain adaptive sliding mode control methods have been developed to reduce chattering [24, 21, 15], and have been implemented on a CDPR [20]. 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. New control methods mixing linear and sliding mode algorithms have been developed to achieve both reduced chattering and

low energy consumption compared to pure sliding mode controllers [23]. The controller then balances between the linear and sliding-mode control types to get a good trade-off between robustness and smoothness of the control input. 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 in spite of perturbations and uncertainties.

The selection of a control strategy depends on the information available on the system. The objective in this paper is then to formulate a recommendation on the control scheme choice for an industrial CDPR, with a focus on the ease of tuning and implementation. As such, in this paper three scenarios are considered along with the corresponding hypotheses on the knowledge of the carried mass. The information on the carried mass is progressively degraded all along the paper to observe the impact on the performance of applicable control strategies:

- First scenario: force sensors are available to measure cable tensions and estimate the mass of the moving parts;
- Second scenario: the mass of the MP is known, but the mass of the metal plate is not known;
- Third scenario: no information is available on the mass of the moving parts.

Several control schemes are experimentally implemented and compared in this paper depending on the provided mass information: a minimal control scheme CS1 with no information on the carried masses, a control scheme CS2 which includes a feedforward term for the compensation of the MP mass, and a control scheme CS3 taking advantage of the robot force sensors to estimate the total mass carried by the robot

to update the feedforward term in real-time. In each scenario, the according control scheme is experimented with both PD and SML control laws and results are compared along the test trajectory. Additionally, a fourth control scheme CS4, based on the SML controller and implementing a cable elasticity compensation term, is presented and experimented in this paper in order to further improve the control robustness and MP pose accuracy.

The paper is organized as follows. First, the prototype and its modeling are introduced in Section 2. Section 3 describes the experimental setup, including the test trajectory and scenarios. The control schemes and controllers are presented in Section 4 and the experimental results are then analyzed in Section 5. Section 6 describes the implementation of a control scheme integrating a cable elasticity compensation term to improve the accuracy of the CDPR. Finally, conclusions are drawn on the recommendation of control schemes and control laws depending on the CDPR equipment and the information on the moving parts.

2. Prototype and modeling

This section deals with the description and modeling of the CDPR prototype, named CAROCA [7] (see Figure 2) used for the experimental comparison of the control schemes.

2.1. CAROCA prototype and ROCKET project

CAROCA is a reconfigurable CDPR prototype developed at IRT Jules Verne, Nantes, France, and is dedicated to industrial operations. A video of a logistics application on the prototype is available². In this paper, its application is the displacement of metal plates of highly variable shape and mass, up to 700 kg, the control objective being to get good accuracy (≈ 1 cm) and repeatability.

²CDPR logistics application at IRT Jules Verne (YouTube): <https://bit.ly/irtjvlogisticscdpr>

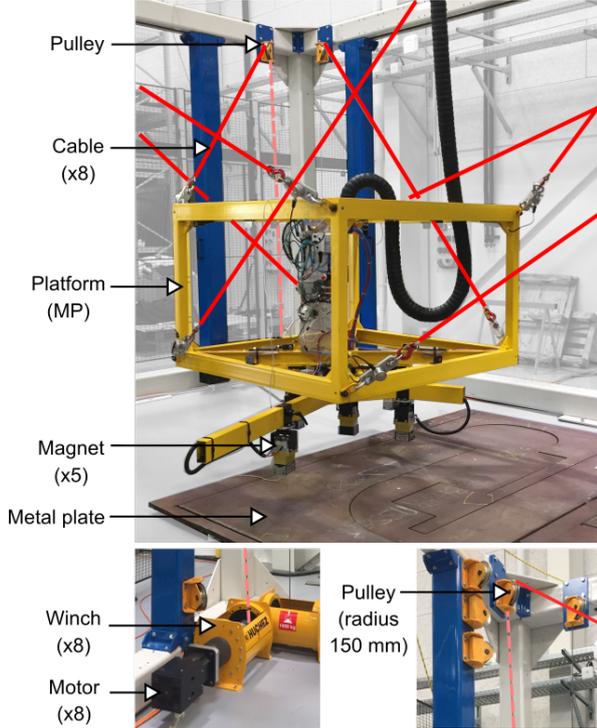


Figure 2: The moving-platform (MP) (mass = 366 kg) equipped with five magnets to handle the metal plates.

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 HuchezTM winches, that are pulling a MP. The winches are actuated by B&R AutomationTM 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 mounted between each motor and each winch. As a consequence, the prototype can lift up to 1 ton. Figure 2 presents the MP of size 1.5 m×1.5 m×1 m and mass 366 kg. Five magnets are embedded under the MP to pick and place metal parts. The robot is also

equipped with TractelTM force sensors located between the cables and the anchor points of the MP (Figure 2). Hardware such as motors and control bay are standard industrial components commercialized by B&R AutomationTM. The robot programming is done under Automation Studio 4.7TM and the whole robot code, from trajectory generation to the controllers implementation, runs on a X20CP3586 industrial CPU (Intel Atom 1.6 GHz processor, 512MB DDR2 SDRAM) in a 2 ms real-time loop (500 Hz).

2.2. Inverse Geometric Model (IGM)

Figure 3 depicts the main geometric parameters of a CDPR and its i^{th} loop-closure equation, $i \in \{1, \dots, m\}$ with m the number of cables attached to the MP. P being the MP geometric center, \mathcal{F}_b is the robot base frame and \mathcal{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 \mathcal{F}_b . Vector ${}^p\mathbf{b}_i$ points from P to B_i and is expressed in frame \mathcal{F}_p . Vector ${}^b\mathbf{p}$ is the position vector of point P expressed in \mathcal{F}_b .

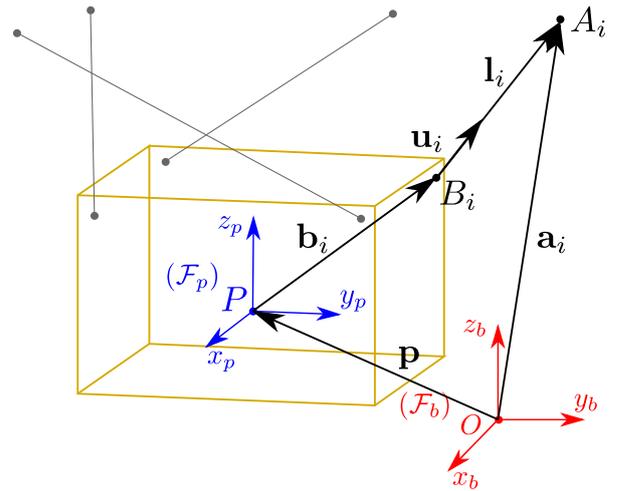


Figure 3: CDPR geometric parameterization, with MP in yellow.

Vector \mathbf{l}_i represents the i^{th} cable vector and

points from B_i to A_i , and reads as

$${}^b\mathbf{l}_i = l_i {}^b\mathbf{u}_i = {}^b\mathbf{a}_i - {}^b\mathbf{p} - {}^b\mathbf{R}_p {}^p\mathbf{b}_i \quad (1)$$

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

$$l_i = \|{}^b\mathbf{l}_i\|_2 \quad {}^b\mathbf{u}_i = \frac{{}^b\mathbf{l}_i}{l_i} \quad (2)$$

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

It should be noted that the CDPR pulleys can be included in the geometric model to improve its accuracy [6, 17].

2.3. Static equilibrium

The static equilibrium of the MP is given by

$$\mathbf{W}\mathbf{t} + \mathbf{w}_e + \mathbf{w}_g = 0 \quad (3)$$

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

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

\mathbf{t} is the cable tension vector, \mathbf{w}_g the wrench applied to the MP due to gravity and \mathbf{w}_e an external wrench expressed in frame \mathcal{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:

$$\mathbf{A}\mathbf{v} = \dot{\mathbf{i}} = \frac{r_w}{R}\dot{\mathbf{q}} \quad \text{with} \quad \mathbf{v} = [{}^b\mathbf{p} \quad {}^b\boldsymbol{\omega}]^T \quad (5)$$

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

$$\mathbf{W} = -\mathbf{A}^T \quad (6)$$

2.5. Dynamic model

From [7], the dynamic model of the CDPR reads as

$$\mathbf{W}\mathbf{t} - \mathbb{I}_p \dot{\mathbf{v}} - \mathbf{C}\mathbf{v} + \mathbf{w}_e + \mathbf{w}_g = 0 \quad (7)$$

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

If G , the center of mass of the MP, does not coincide with P , the wrench \mathbf{w}_g due to the gravity acceleration \mathbf{g} is defined as

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

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

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

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

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

$$\mathbf{I}_p = {}^b\mathbf{R}_p \mathbf{I}_g {}^b\mathbf{R}_p^T - \frac{\mathbf{M}\hat{\mathbf{S}}_p \mathbf{M}\hat{\mathbf{S}}_p}{m_p} \quad (10)$$

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

$$\mathbf{C}\mathbf{v} = \begin{bmatrix} {}^b\hat{\boldsymbol{\omega}} {}^b\boldsymbol{\omega} \mathbf{M}\hat{\mathbf{S}}_p \\ {}^b\hat{\boldsymbol{\omega}} \mathbf{I}_p {}^b\boldsymbol{\omega} \end{bmatrix} \quad (11)$$

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

2.6. State system

The dynamics of the motors [13] are given by

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

where \mathbf{I}_q is the diagonal matrix containing the moment of inertia of the gearmotors and winches, 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 (12), the CDPR inverse kinematic model (5) and the CDPR dynamic model (7), defining the state vector as $\mathbf{x} = [\mathbf{q} \ \dot{\mathbf{q}}]^T$ and the system input as $\mathbf{u} = \boldsymbol{\tau}_m$, the system can be represented as a standard nonlinear system of the form

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

Notice that the system is affine in the control input \mathbf{u} . Furthermore, $f(\mathbf{x})$ is uncertain due to the relationship between cable tensions \mathbf{t} and both the external wrench \mathbf{w}_e and the wrench due to gravity \mathbf{w}_g in (7), which is a function of the payload.

3. Test trajectory and experimental setup

Experimental tests are realized on the CAROCA prototype along a test trajectory with several loads.

Three scenarios are considered depending on the level of information on the total carried mass, namely the masses of the MP and the eventual metal plate. The scenarios are ordered starting from the most accurate information on the masses; then the mass information is degraded in order to evaluate the impact on the control performances. In each scenario, an appropriate control scheme must be implemented, based on the available system information.

3.1. Scenarios and hypotheses

The three scenarios are defined as follows:

- Scenario A (“Cable tensions measurements”): The CDPR is equipped with force sensors to measure cable tensions and estimate the carried masses in real-time

- Scenario B (“Partial information”): No sensors are available, but MP mass is known
- Scenario C (“No information”): No information on both the MP and metal plate masses

3.2. Test trajectory

The test trajectory generated to evaluate the performance of the different control methods describes a typical pick-and-place application. The trajectory is generated using s-curves, that ensure continuous velocity and acceleration trajectory profiles [2]. The x -axis of the frame \mathcal{F}_b is defined along the width of the CDPR, the y -axis along its length and the z -axis along its height. The test trajectory consists of (see Figure 4):

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;
5. Then the trajectory is repeated in the opposite direction, from point E to point A.

The total test trajectory time is 60 s, 30 s for the MP to move from point A to point E and 30 s for the MP to move from point E to point A.

3.3. Masses

In each scenario, the test is first performed on the CDPR with the empty moving-platform of mass 366 kg. Then, in order to evaluate the control robustness, two metal plates are successively carried by the MP. Three cases are then considered:

1. the empty moving-platform of mass 366 kg (MP);

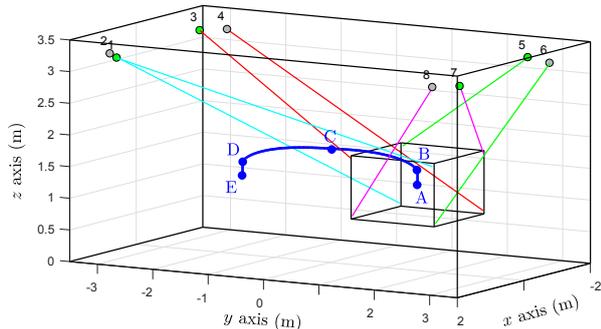


Figure 4: Test trajectory (blue) and cable arrangement of the CDPR.

2. the MP and a metal plate M1 of mass 122 kg, for a total mass equal to 488 kg (MPM1);
3. the MP and a metal plate M2 of mass 249 kg, for a total mass equal to 615 kg (MPM2)

Note that the variation in the payload is significant, namely +33% (M1) and +68% (M2) with respect to the MP mass, respectively. Figure 5 presents the MP carrying the metal plates M1 and M2.

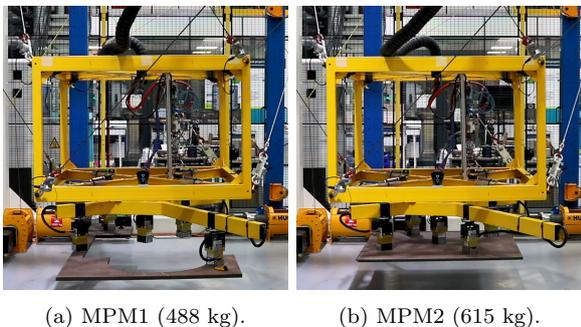


Figure 5: MP carrying a metal plate: (a) M1, (b) M2.

4. Control

As presented in Section 1, the main objective is the development of a robot for pick-and-place operations of metal plates of different masses and shapes in an industrial environment. The stability of the closed-loop system, its robustness, and repeatability despite the load changes

are searched for. The developed control strategies should also be compatible with real-time implementation. The resolution of the direct geometric model to estimate the MP pose could lead to longer computation time and the risk of discontinuities in the solution found by the optimization method. A simpler approach implementing the control law in the joint space is preferred for this particular application.

4.1. Control schemes

The considered control schemes are presented in what follows. Due to the parallel nature of CDPRs, all signals are treated as vectors in the following control schemes. Each actuation chain from the motor to the cable has its own controller. For a fair comparison in the experimental results, the controllers are tuned once and the same parameters are used in all control schemes.

4.1.1. CS1: Basic control scheme

The first control scheme implemented, denoted as CS1, is presented in Figure 6. The *controller* box at the center of the scheme is equal to either the PD controller or SML controller presented in the sequel (Section 4.2). The corresponding control scheme is then referenced as CS1-PD or CS1-SML accordingly.

In the control scheme CS1 (Figure 6) \mathbf{s}_d is the 6-dimensional vector containing the desired Cartesian position and orientation of the MP, and \mathbf{v}_d the desired MP twist (linear and angular MP velocities). Since the CAROCA prototype has eight cables, \mathbf{q}_d , $\dot{\mathbf{q}}_d$ and $\boldsymbol{\tau}_m$ are the 8×1 vectors of desired motor angular positions, desired velocities and applied torques, respectively. In this first scheme, the desired motor torque vector $\boldsymbol{\tau}_m$ is the output of the controller.

The test trajectory described in Section 3 provides the desired MP pose, *i.e.* MP position and orientation and MP twist, which are converted into desired motor positions and velocities using the inverse CDPR geometric and kinematic models, from (1) and (5).

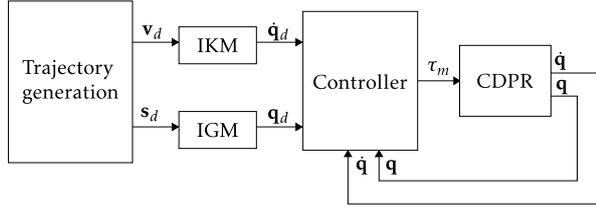


Figure 6: CS1 control scheme.

Since the control scheme CS1 is very simple, the selected controller will have to be robust in order to achieve the desired performances.

4.1.2. CS2: Control scheme with feedforward term

The goal of the control scheme CS2 is to compute in advance the necessary motor torques ensuring the dynamic equilibrium of the MP along the test trajectory, in order to improve control robustness towards variations and uncertainties in the payload. Figure 7 presents the CS2 control scheme with feedforward, where $\dot{\mathbf{v}}_d$ contains the linear and angular accelerations of the MP. $\boldsymbol{\tau}_c$ is a 8×1 vector provided by the controller.

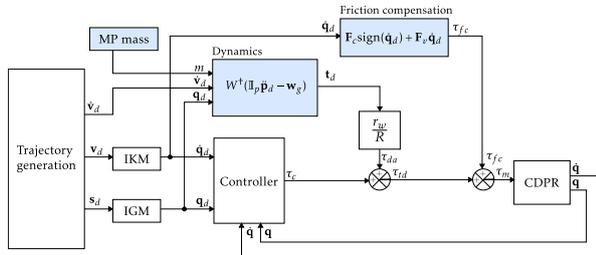


Figure 7: CS2 control scheme with feedforward terms.

Based on the dynamic model of the robot, it is possible to add a term to the controller that anticipates the MP dynamics and compensates for a given mass value along the trajectory. From Eq. (7), this feedforward term is defined as

$$\tau_{da} = \frac{r_w \mathbf{W}^\dagger (\mathbb{I}_p \dot{\mathbf{v}}_d - \mathbf{w}_g)}{R} \quad (14)$$

The term \mathbf{w}_g depends on the load mass. However, for the metal plate handling appli-

cation, the metal plate masses are considered unknown. As a result, only the mass of the MP m_{MP} is considered in the feedforward term (366 kg) [17]:

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

with x_G and y_G the displacement of the center of mass G along the corresponding axes. Since m_{MP} is the mass of the MP, it equals 366 kg for all payloads (MP, MPM1 and MPM2 - see Section 3.3). This introduces an error on purpose in the latter two cases, that is when metal plate M_1 or M_2 is carried, in order to quantify the control robustness towards the mass variation. In the sequel, x_G and y_G are equal to zero since the carried metal plate mass is not known, but the feedforward term can be improved if this information is available.

A linear friction model [10] has been implemented in each actuation chain to compensate for the loss in the motors, gearbox, and winches:

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

with $\boldsymbol{\tau}_{fc}$ the friction compensation and $\dot{\mathbf{q}}_d$ the desired motor rate vector. \mathbf{F}_c and \mathbf{F}_v are respectively the vectors containing the static and viscous friction coefficients separately identified for each actuator. This architecture can also reduce chattering and overshooting by allowing for smaller gains in both controllers, resulting in a smoother behavior of the robot.

The notation CS2⁺ will be used in the particular case where the total mass is known, that is the mass of both the MP and the metal plates. In this case, the term m_{MP} should be replaced by m_{tot} , which contains the total load mass. This notation is used in the experimental section Section 6 on experimental results, where the benefits of a cable elasticity compensation term are studied in the most favorable control conditions.

4.1.3. CS3: Control scheme with feedforward term using real-time mass estimation

The CAROCA prototype is equipped with force sensors located between the cable and the MP anchor points. Thus, cable tensions can be directly measured and leveraged to compute the total mass of the set composed of the MP and a metal plate. A control scheme can then be designed using the mass estimation, which updates the value of the mass in the feedforward term [17]. The third control scheme, named CS3, with real-time update of the feedforward term τ_{idm} is depicted in Figure 8. In an almost static case where the MP dynamics are negligible, (7) can be rewritten as:

$$\mathbf{W}\mathbf{t} + \mathbf{w}_g = 0 \quad (17)$$

From (8) and (17), assuming that the MP orientation remains constant and null [18], \mathbf{w}_g can be expressed as:

$$\mathbf{w}_g = \begin{bmatrix} 0 \\ 0 \\ -mg \\ -mgy_G \\ mgx_G \\ 0 \end{bmatrix} = -\mathbf{W}\mathbf{t} = -\mathbf{W} \begin{bmatrix} t_1 \\ \vdots \\ t_8 \end{bmatrix} \quad (18)$$

Knowing the cable tensions \mathbf{t} and the wrench matrix \mathbf{W} , it is then possible to calculate the payload mass m and the Cartesian coordinates x_G and y_G of the center of mass of the set composed of the MP and the carried payload, expressed in frame \mathcal{F}_p . With m_{RT} the real-time estimation of the mass of the total load, namely the MP and the carried metal plate, the vector \mathbf{w}_g in the feedforward term equals:

$$\mathbf{w}_g = [0 \quad 0 \quad -m_{RT}g \quad -m_{RT}gy_G \quad m_{RT}gx_G \quad 0]^T \quad (19)$$

Notice that the center of mass displacements x_G and y_G can also be estimated in real-time from (18). For a fair comparison of the mass compensation between CS2 and CS3 in the experimental results, x_G and y_G are also equal to zero in CS3.

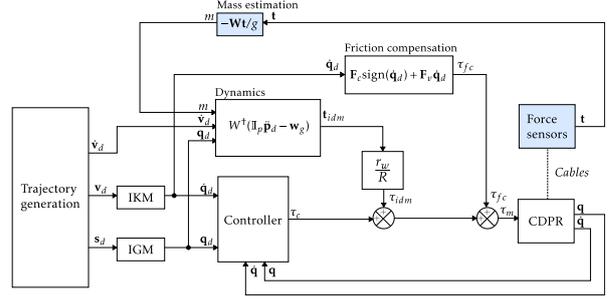


Figure 8: CS3 control scheme with real-time mass compensation.

4.2. Controllers

This section describes the controllers designed for the *Controller* box. Recall that a similar controller is applied to the eight motors of the CDPR. Two families of controllers have been tested: (i) proportional-derivative and (ii) new generation of sliding mode control.

4.2.1. Proportional-derivative controller

The control input vector of motor torques $\boldsymbol{\tau}_{PD}$ generated by the controller reads as

$$\boldsymbol{\tau}_c = \mathbf{K}_p \mathbf{e}_q + \mathbf{K}_d \mathbf{e}_{\dot{q}} \quad (20)$$

with \mathbf{e}_q the difference between the desired and actual motor positions and $\mathbf{e}_{\dot{q}}$ being the difference between the desired and actual motors velocities. In a decentralized control scheme, each motor is independently controlled: then, the matrices \mathbf{K}_p and \mathbf{K}_d are diagonal and read as

$$\mathbf{K}_p = \text{diag}(K_{p,1} \dots K_{p,8}) \quad \text{and} \quad \mathbf{K}_d = \text{diag}(K_{d,1} \dots K_{d,8}) \quad (21)$$

For simplicity and since the identified motor friction coefficients have been found to be similar across all motors, the eight decentralized controllers have been tuned in a similar manner: $K_p = K_{p,1} = K_{p,2} \dots$ and $K_d = K_{d,1} = K_{d,2} \dots$. K_p and K_d have been tuned so as the robot achieves satisfactory accuracy and stability with a lighter version of MP of mass equal to

300 kg, without external wrench or load. The values K_p and K_d for the experiments are given in Table 1.

The PD controller has been tuned using the standard Ziegler-Nichols method [28]. Values of the proportional and derivative gains, K_p and K_d respectively, are then derived from Table 1, K_u being the “critical” gain producing oscillations of period T_u .

Table 1: Ziegler-Nichols equations for PD control and chosen values.

Gain	Ziegler-Nichols (PD)	Values
K_p	$0.8K_u$	0.3
K_d	$K_u T_u / 10$	0.03

4.2.2. Balancing sliding mode/linear controller

The main feature of this class of controllers [23] is that it balances between linear and sliding mode algorithms to take advantage of these two classes of control, that are robustness and accuracy for sliding mode control, and smoothness and reduced energy consumption for linear state feedback. Similarly to the PD controller, eight individual SML controllers have been implemented. The SML control law is formulated in the following. Define the sliding vector σ as

$$\sigma = (\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \lambda(\mathbf{q}_d - \mathbf{q}) \quad (22)$$

$$= e_{\dot{\mathbf{q}}} + \lambda e_{\mathbf{q}} \quad (23)$$

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, $e_{\dot{\mathbf{q}}}$ and $e_{\mathbf{q}}$ the corresponding tracking errors and λ a strictly positive parameter ($\lambda > 0$).

Sliding mode control must ensure that the sliding vector reaches and is maintained at zero in a finite time [24, 21]: given the definition (22) of σ , when the sliding variable 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 dynamics of the closed-loop system is defined by the differential equations $e_{\dot{q},i} = -\lambda e_{q,i}$, as such the higher λ , the faster the convergence. This is described as the *sliding phase*.

σ has a relative degree of one with respect to τ_m . As a consequence, the time derivative of the sliding variable reads as

$$\dot{\sigma} = e_{\dot{\mathbf{q}}} + \lambda e_{\dot{\mathbf{q}}} \quad (24)$$

$$= (\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}) + \lambda(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \quad (25)$$

$$= \ddot{\mathbf{q}}_d + \lambda(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) - \ddot{\mathbf{q}} \quad (26)$$

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

$$\dot{\sigma} = \mathbf{A}_\sigma(\dot{\mathbf{q}}, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, \mathbf{t}) + \mathbf{B}_\sigma \tau_m \quad (27)$$

with $\mathbf{B}_\sigma = \mathbf{I}_q^{-1}$. Notice that this latter matrix is a diagonal one that makes the system as a decoupled one.

Applying standard sliding mode control could induce chattering on the input. A solution is the application of second-order sliding mode control that ensures $\sigma = \dot{\sigma} = \mathbf{0}$. Among the second-order sliding mode algorithms, the choice in the sequel is the twisting controller [15, 21]. In order to design control law, the sliding vector is derived an additional time that gives

$$\ddot{\sigma} = (\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}) + \lambda(\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}) \quad (28)$$

$$= \ddot{\mathbf{q}}_d + \lambda(\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}) - \ddot{\mathbf{q}} \quad (29)$$

$$= \mathbf{H}_\sigma(\ddot{\mathbf{q}}, \ddot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, \mathbf{t}) + \mathbf{J}_\sigma \dot{\tau}_m \quad (30)$$

with $\mathbf{J}_\sigma = \mathbf{B}_\sigma$. Each component of $\dot{\tau}_m$ [22] is defined as

$$\dot{\tau}_{m,i} = -K_1[\sigma_i]^{\frac{\alpha_i}{2-\alpha_i}} - K_2[\dot{\sigma}_i]^{\alpha_i} \quad (31)$$

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

$$[\sigma_i]^{\alpha_i} = |\sigma_i|^{\alpha_i} \text{sign}(\sigma_i) \quad (32)$$

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

$$\alpha_i = \max \left(-\beta \left(\frac{|\sigma_i|}{|\sigma_i| + \epsilon_\sigma} + \frac{|\dot{\sigma}_i|}{|\dot{\sigma}_i| + \epsilon_{\dot{\sigma}}} \right) + 1, 0 \right) \quad (33)$$

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 $\tau_{m,i}$ is then obtained by integrating (31).

The principle of the SML controller (31)-(33) is the following: the value of the variable α_i 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, $\alpha_i \rightarrow 0$; from (31), the control becomes a twisting one [15]

$$\dot{\tau}_{m,i} = -K_1 \text{sign}(\sigma_i) - K_2 \text{sign}(\dot{\sigma}_i) \quad (34)$$

that ensures, in practice, the convergence of σ_i and $\dot{\sigma}_i$ to a vicinity of $(0, 0)$ in a 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 $\alpha_i \rightarrow 1$. α_i regulates the trade-off between accuracy and chattering reduction. $\dot{\tau}_{m,i}$ then tends towards the expression

$$\dot{\tau}_{m,i} = -K_1 \sigma_i - K_2 \dot{\sigma}_i \quad (35)$$

In order to guarantee convergence of the closed-loop system, the gains K_1 and K_2 must be positive and follow the condition [15]

$$\begin{aligned} K_1 &> K_2 > 0, \\ (K_1 - K_2)j_m &> h_M \\ (K_1 + K_2)j_m - h_M &> (K_1 - K_2)j_m + h_M \end{aligned} \quad (36)$$

with h_M , j_m and j_M positive constants such that for each motor and from (27)

$$|h_{\sigma,i}(\cdot)| \leq h_M \quad \text{and} \quad 0 < j_m \leq j_{\sigma,i,i} \leq j_M \quad (37)$$

with $h_{\sigma,i}$ the i^{th} -component of vector \mathbf{H}_σ and $j_{\sigma,i,i}$ the (i, i) -component of matrix \mathbf{J}_σ . The reduced energy consumption of the SML controller compared to the twisting algorithm is ensured by the following condition [22]:

$$K_1 \epsilon_\sigma + K_2 \epsilon_{\dot{\sigma}} < K_1 - K_2 \quad (38)$$

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. Notice that, in a sake of simplicity, both the gains have been tuned at similar values for the eight motors. It could be interesting to individually tune each controller if the efforts applied to the motors are significantly unbalanced, for example due to a specific robot geometry or a large displacement of the moving parts center of mass with respect to the MP center. Furthermore, β and $\epsilon_\sigma, \epsilon_{\dot{\sigma}}$ 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 $\epsilon_\sigma, \epsilon_{\dot{\sigma}}$, 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 $\epsilon_\sigma, \epsilon_{\dot{\sigma}}$ 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. $\lambda > 0$ is chosen 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 σ and $\dot{\sigma}$ values, choose values for ϵ_σ and $\epsilon_{\dot{\sigma}}$ with the same ratio as between σ and $\dot{\sigma}$;
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 end up with a good compromise between precision 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
Value	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 $e_{\ddot{q}}$ appearing in $\dot{\sigma}$ are derived from the motor velocity errors with the usual Euler method.

5. Experimental results

This section covers the experimental results obtained in the three scenarios introduced in Section 3. In each scenario, an applicable control scheme (CS1, CS2 or CS3) has been implemented on the prototype in combination with the PD or SML controller. The performance of each control strategy is analyzed for the three loads (MP, MPM1 and MPM2).

The applicable control schemes in each scenario are:

- Control schemes CS3-PD and CS3-SML, with real-time mass estimation in Scenario A
- Control schemes CS2-PD and CS2-SML, with MP mass compensation in Scenario B
- Control schemes CS1-PD and CS1-SML, with basic architecture in Scenario C

5.1. Motor position errors

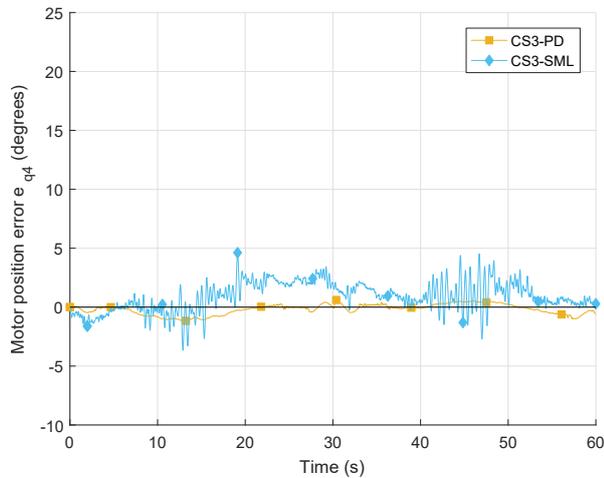
First, the position errors in the joint space are observed. Due to lack of space, only the motor position error of a single motor, Motor 4, is plotted versus time in each scenario. The *Root Mean Square* (RMS) of all motor errors along the second-half of the test trajectory (travel E to A from $t = 30$ s to $t = 60$ s) are also provided to assess the overall performance of the seven other motors, once the controllers are in steady-state.

5.1.1. Scenario A (“Cable tensions measurements”): Force sensors are available

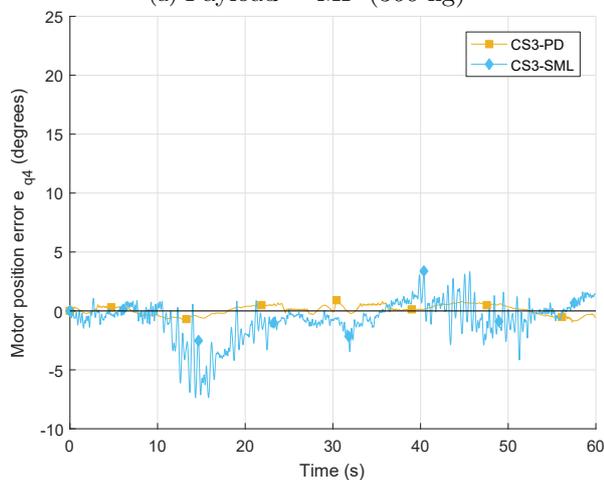
If the robot is equipped with force sensors, the most refined control scheme CS3 can be implemented, including the real-time mass estimation and the feedforward term update.

The motor position errors of Motor 4 obtained with CS3-PD and CS3-SML along the test trajectory are presented in Figure 9. It is apparent that the error with CS3-PD is consistently small for all loads with an error smaller than 2 degrees. On the other hand, CS3-SML suffers from oscillations with a frequency around 1 Hz, the amplitude of the variations increasing with the load. Additionally, the maximum error of Motor 4 reaches up to 14 degrees with CS3-SML.

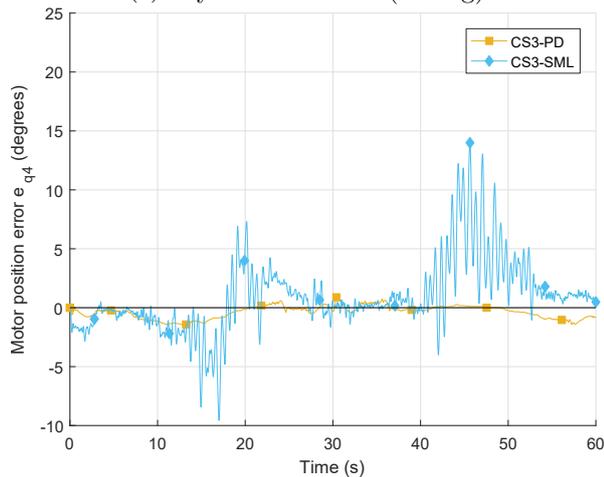
The oscillations observed with CS3-SML are due to the combination of the real-time mass estimation with the sliding mode control chattering. The mass estimation for the three loads are plotted in Figure 10, for both CS3-PD and CS3-SML, along the test trajectory. Variations in the mass estimation for CS3-SML can be observed, while the mass estimation is much more stable with CS3-PD. Chattering on the motor torque might induce vibrations in the cables, leading to cable tensions variations, which then impact the mass estimation and the feedforward term. [As a consequence, it appears that the simplified real-time mass estimation method proposed in this paper is not well suited as a direct feedback in the case](#)



(a) Payload = MP (366 kg)



(b) Payload = MPM1 (488 kg)



(c) Payload = MPM2 (615 kg)

Figure 9: Motor 4 position errors along the test trajectory for CS3 control scheme, with PD and SML controllers.

of CS3-SML. In this case, it could be interesting to implement a higher level momentum-based payload estimation method that takes into account the MP acceleration in the calculation [8], in order to achieve a more stable payload estimation in spite of MP oscillations.

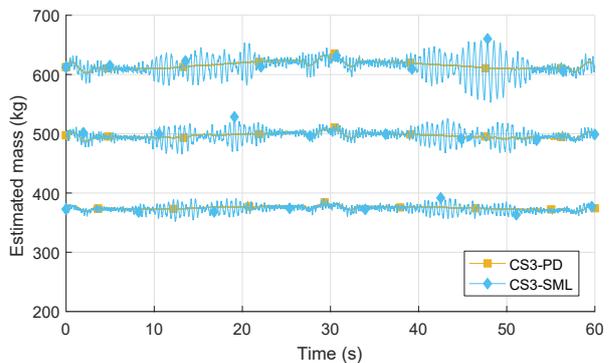


Figure 10: Estimated mass (kg) for the three loads (bottom. MP; middle. MPM1; top. MPM2) with CS3-PD and CS3-SML control schemes along the test trajectory.

Figure 11 contains the RMS of all motor errors along the second-half of the test trajectory, when the controllers are in steady state. Again, CS3-SML leads to larger values with a slight increase with the load. On the other hand, the eight motors present a similar level of performances across all motors with CS3-PD for the three loads although there are still some small differences.

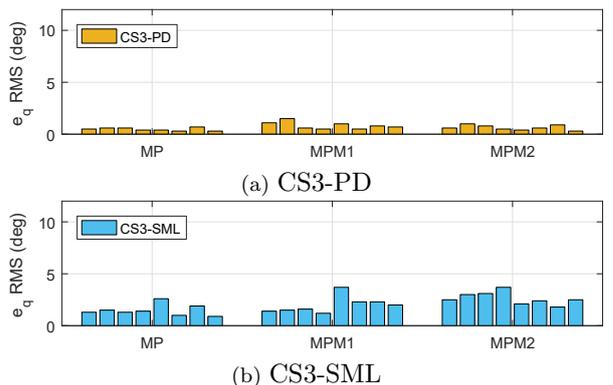


Figure 11: RMS of e_q in degrees for Motors 1 to 8 (left to right), with CS3 control scheme along the the second half of the test trajectory (degrees).

To conclude on scenario A, it appears that

the PD controller is more appropriate than the SML inside of the proposed CS3 architecture.

5.1.2. Scenario B (“Partial information”): No force sensors, but MP mass is known

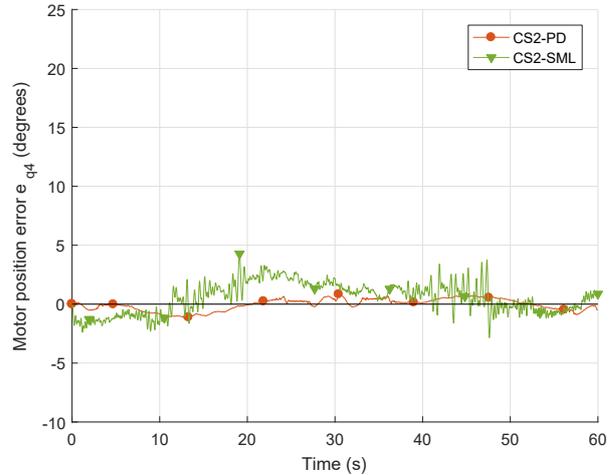
In Scenario B, the robot is not equipped with force sensors, which means that control scheme CS3 cannot be implemented, but the mass of the MP alone is considered known. The feedforward term can then be computed from this theoretical value: the control scheme CS2 is used in this scenario, with the feedforward term computed from the mass of the single MP equal to 366 kg, without taking into account the possible carried metal plate.

Motor 4 position errors along the test trajectory are plotted in Figure 12. For CS2-PD, no static error is visible when the motors only support the MP, whose mass is compensated by the feedforward term. With load MP, the performance level of CS2-PD is close to the one obtained with CS3-PD. However, it can be observed that a static error appears when the MP carries a metal plate M1 or M2, since its mass is unknown.

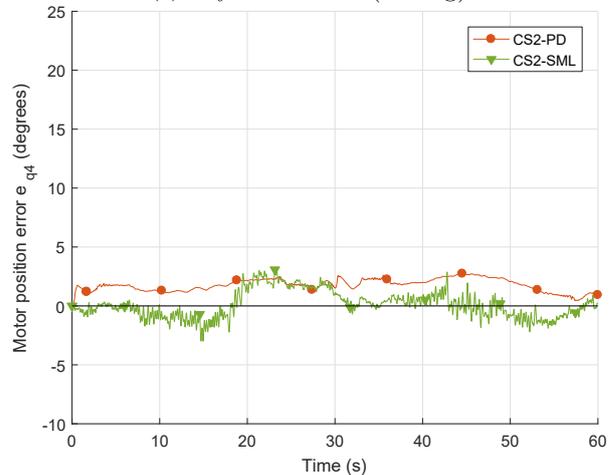
For CS2-SML, motor vibrations are still visible, as well as some variations in the shape of the error curve, but the general behavior is now constant for all loads. This is due to the intrinsic robustness of the controller, confirmed by the fact that CS2-SML motor position error is lower than for CS2-PD for loads MPM1 and MPM2. The only exception where CS2-PD fares slightly better than CS2-SML is in the case of the single MP, where the mass is accurately compensated for by the feedforward.

Looking at the RMS in Figure 13, it is clear that CS2-PD provides the best results for all motors with load MP, while the error greatly increases as a metal plate of unknown mass is carried. On the other hand, the CS2-SML control architecture provides similar performances no matter the carried load.

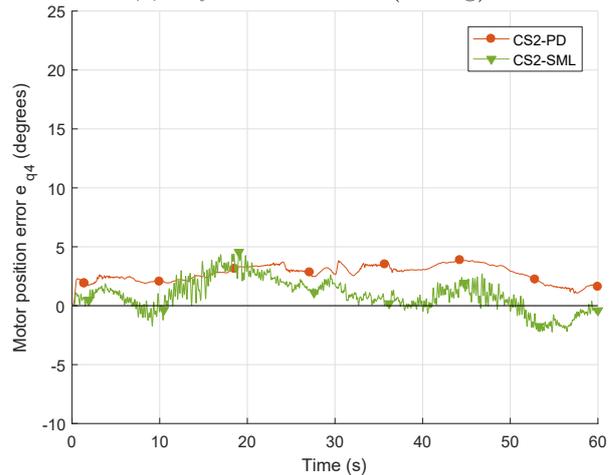
Since the CS2 control scheme includes a feedforward term to compensate only for the MP mass, it is interesting to observe its contribu-



(a) Payload = MP (366 kg)



(b) Payload = MPM1 (488 kg)



(c) Payload = MPM2 (615 kg)

Figure 12: Motor 4 position errors along the test trajectory for CS2 control scheme, with PD and SML controllers

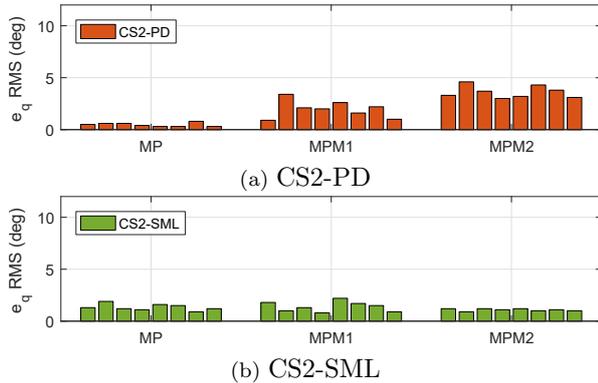


Figure 13: RMS of e_q in degrees for Motors 1 to 8 (left to right), with CS2 along the the second half of the test trajectory (degrees).

tion to the total motor torque. The torque control input of Motor 4, $\tau_{m,4}$, is shown in Figure 14, in the case of the heaviest payload (MPM2) for the two control strategies, CS2-PD and CS2-SML. The contribution of the feedforward terms for Motor 4, $\tau_{ff,4}$ corresponding to the sum of the dynamic anticipation and friction compensation is also plotted.

Recall that the feedforward applied in the CS2 architecture is computed based on the mass of the MP only whose mass equals 366 kg. It appears that the feedforward term is not sufficient to generate the total torque necessary to carry MPM2 load, but still represents the majority of the motor torque. In particular, at the start of the test trajectory the feedforward term generates most of the initial starting torque of 2.2 N m. Each controller then has to compute the remaining necessary torque to achieve the desired position to compensate for the modeling inaccuracies. The similarity between the shapes of the feedforward term and measured torques confirms that a good anticipation is possible with the current modeling of the robot, the difference being mostly due to the assumption on the carried mass.

From Scenario B, it can be concluded that the CS2-PD control architecture is interesting when the total theoretical mass is known, in which case the performances are very close

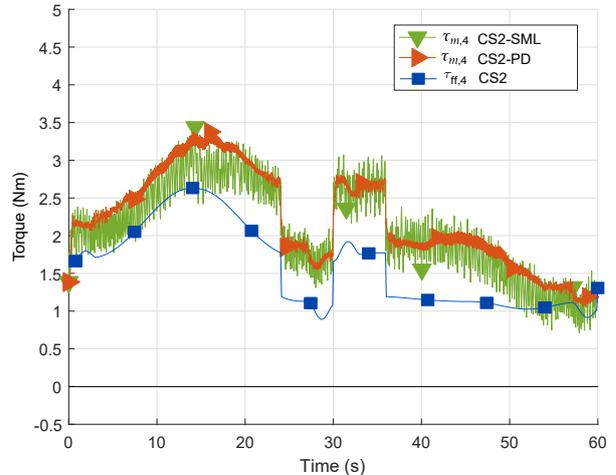


Figure 14: Applied motor torques for Motor 4 $\tau_{m,4}$ and load MPM2 for both controllers, and CS2 feedforward term, $\tau_{ff,4}$.

to those obtained with CS3-PD based on the force sensors. Focusing on the results obtained with load MP only, this scenario also shows the good performances of the CS2⁺ control scheme, where the feedforward term compensates for the total carried mass: if the total mass information is available, the motor errors obtained will be small with both controllers. However, a static error appears when a metal plate of unknown mass is added, in which case CS2-SML appears to be more robust and interesting possibility for the application when an unknown metal plate is carried.

5.1.3. Scenario C (“No information”): No force sensors and no information on carried mass

The CS1 control scheme is now applied since no information is available on the platform mass. Figure 15 presents Motor 4 angular error e_{q4} along the test trajectory, with the three loads. As expected, CS1-PD presents static error for all loads, which increases with the mass. The maximum error of Motor 4 reaches 10 degrees for the heaviest payload (MPM2).

In the case of CS1-SML, the motor position error oscillates more and a transient appears, especially for the heaviest loads. However, it

should be noted that in spite the large transient, this error is always converging to zero by the end of the first part of the test trajectory ($t = 30\text{ s}$), for all masses. In continuous operation, the CS1-SML controller then maintain the performances of the steady state.

From Figure 16, the increase of the motor position errors RMS along with the mass increase is clearly observed for CS1-PD, in particular for Motor 6 that presents a RMS of 6.2 degrees for load MP, and 10.9 degrees for load MPM2. On the other hand, once in steady state, CS1-SML provides stable values across all criteria and for all load levels, with a maximum of 4.1 degrees (Motor 8). It appears however that for each load, a single motor presents a significantly higher error than the other ones. Since the control scheme CS1 experimented is decentralized, it is expected that a better tuning of the individual controllers can help to balance these errors.

In summary for the CS1 control scheme, CS1-PD leads to an expected static error, while CS1-SML suffers from a strong transient behavior in motor position error, but this error is reduced by an efficient way over the course of the test trajectory. The time taken to reduce the tracking error is determined by λ ; as an example $\lambda = 0.15$ corresponds to a convergence time of around 20 seconds, which is observed in Figure 15(c). This value has been selected low due to the large payload in order to avoid high dynamics of the MP in these experiments, but could have been increased to reduce the duration of the transient phase. However, once in steady state, the sliding mode controller error is consistently lower than with the PD control law.

5.1.4. Evolution of α in PC1-SML controller

The mean value of α along the test trajectory and across all motors with PC1-SML controller is presented in Figure 17. It can be noticed that, with the current tuning of the SML controller, the values of α are relatively low with a mean value equal to 0.12. It means that the

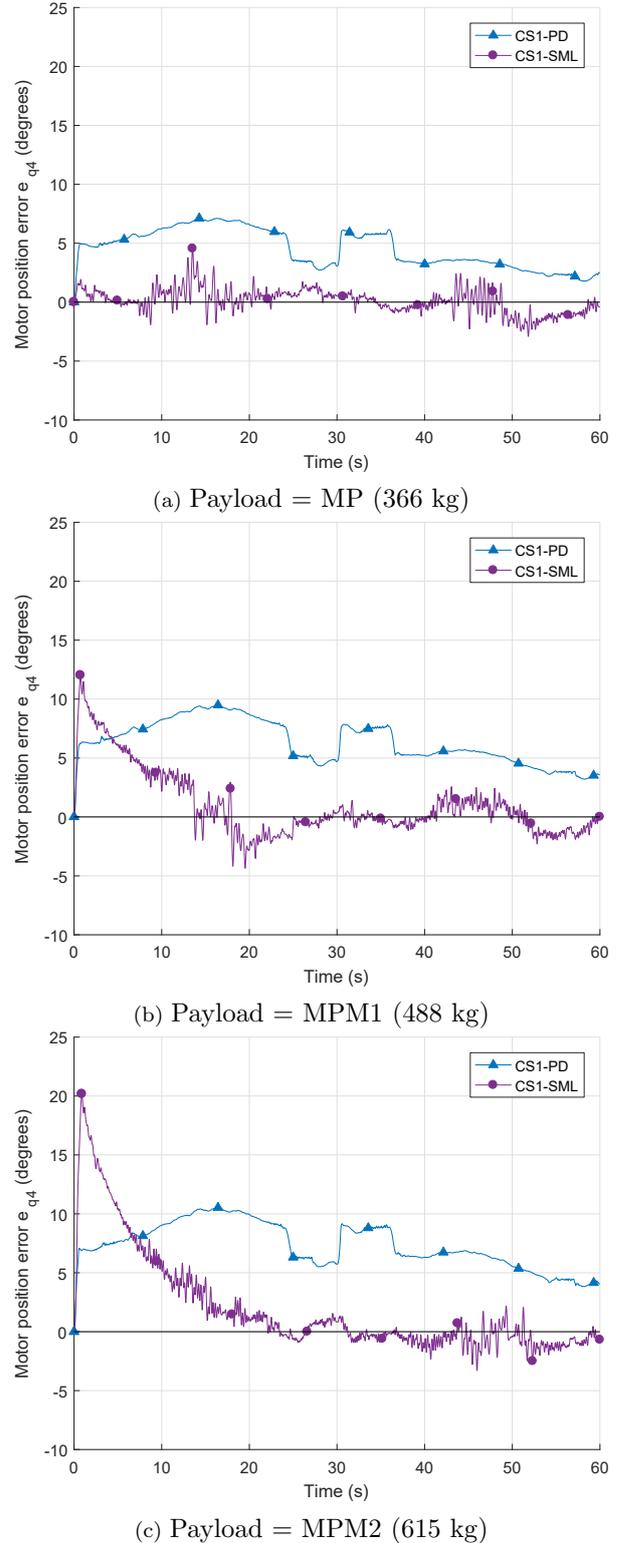


Figure 15: Motor 4 position errors along the test trajectory for CS1 control scheme, with PD and SML controllers.

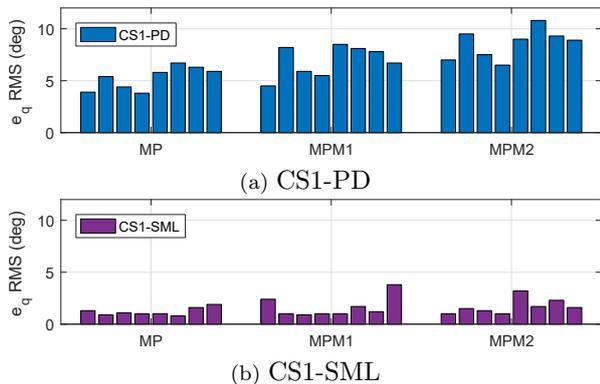


Figure 16: RMS of e_q in degrees for Motors 1 to 8 (left to right), with CS1 along the the second half of the test trajectory (degrees).

controller is mostly a sliding mode one along the test trajectory, due to the fact that system has a high level of uncertainties. As expected, the larger the payload, the closer the controller to sliding mode control *i.e.* the lower α .

Figure 18 shows the evolution of α for Motor 4 for the heaviest load (MPM2), versus time. It is noteworthy that around $t = 7$ s and $t = 27$ s, α reaches higher values for short periods of time. Meanwhile, lower motor position oscillations are shown in Figure 15. Contrarily, at around $t = 15$ s, the lowest values for α lead to higher oscillations. The control smoothness is effectively linked to higher values of α , leaning towards the linear control algorithm.

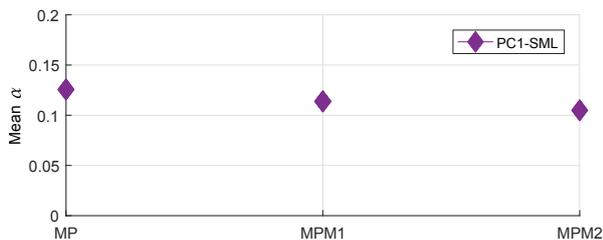


Figure 17: Mean of α along the test trajectory, across all motors and for each load.

5.2. Cartesian errors

During the test trajectory, the MP pose is now captured using an HTC VIVE. A tracker is attached on a side of the MP and is located

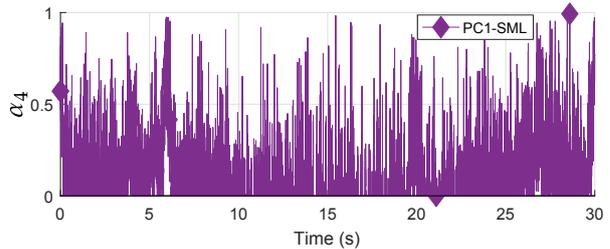


Figure 18: Evolution of α_4 for load MPM2 (615 kg).

inside the workspace using infrared lasers. This tracker has an offset in the base frame compared to the center of the MP, P , equal to $[-0.001 \quad -0.330 \quad 0.695]^T$ m, expressed in \mathcal{F}_b . In the sequel, the platform tracking error in the Cartesian space is calculated at the tracker level. The error is then calculated between the desired and measured tracker poses along the trajectory. The tracking errors of the MP are observed along the axis z_b , given that the vertical motions (along z_b) are the most affected by the load changes.

Figures 19(a)–19(c) present the Cartesian error between the desired and measured position of the tracker along the vertical axis z_b . The overall wave form of the Cartesian tracking error is mainly due to the MP orientation along the trajectory since the tracker is located on a side of the MP, along with CDRP modeling errors and measurement tool inaccuracies. Nonetheless, it is possible to compare the controllers behaviors and accuracy relatively one to another. The downward trend of the average error is, on the other hand, directly due to the increase in mass and cable elasticity.

As previously seen with the motor position error, the CS1 architecture suffers from an initial error, resulting in a drop of the MP. The static error of the CS1-PD architecture, and the initial error of CS1-SML are clearly visible in Figures 19(a)–19(c). On the other hand, CS3-PD and CS2-SML are the most robust architectures. Finally, once the initial transient has passed and steady state is reached CS1-SML provides performances close to the best

working control architecture, which makes it the best alternative if the payload is not well known compared to CS1-PD and CS2-PD.

6. Control scheme with elasticity compensation term

In the previous results, it has been observed that the larger the payload, the lower the position of the MP. This point-displacement of the MP is mainly due to cable elasticity. A method to reduce the MP positioning error is the addition of a cable elasticity compensation loop [11]. Therefore, this section aims to evaluate the effect of such a cable elongation correction onto the performances of the CAROCA prototype for the pick-and-place application.

6.1. Elasticity compensation term

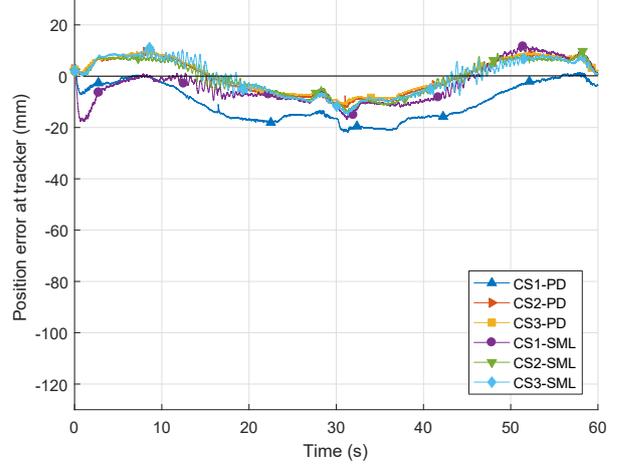
In order to isolate the impact of the elasticity compensation term, a robust control scheme must be selected for the implementation of the cable elongation compensation term.

Control architecture CS2-SML is selected for its good performances without the real-time mass estimation and the robustness of the sliding mode control algorithm versus perturbations. Indeed, the SML controller remains stable even in case of strong perturbations, as previously observed with CS3-SML in scenario A.

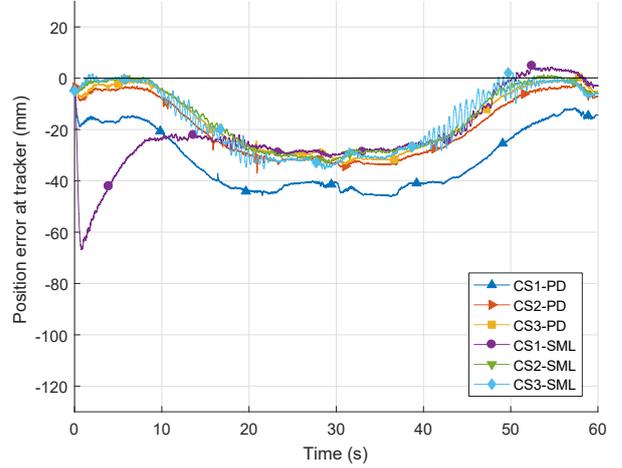
Recall that CS2⁺ is the particular case of CS2-SML in which the mass given to the feed-forward term is equal to the real total mass (either equal to m_{MP} , m_{MPM1} or m_{MPM2} depending on the current load), so that the control architecture is the most accurate possible at the motor level.

The addition of the cable elongation compensation loop to CS2⁺ then makes the new control scheme CS4 (Figure 20).

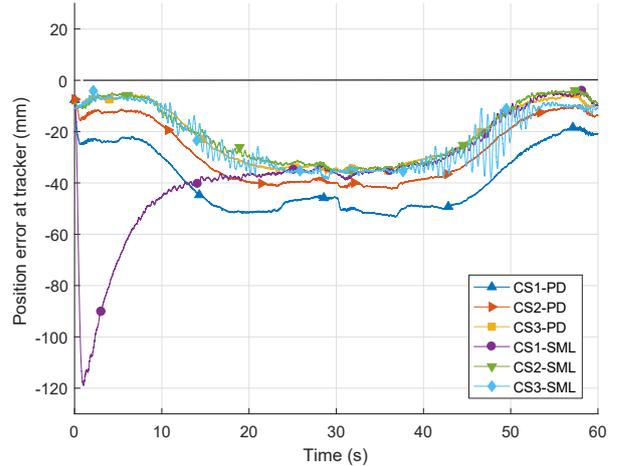
Based on a cable elasticity model and the force sensors readings, it is possible to compute the necessary motor position correction $\delta\mathbf{q}_E$ to compensate for the cable elongation. From a linear cable elasticity model, the cable tensions



(a) Payload = MP (366 kg)



(b) Payload = MPM1 (488 kg)



(c) Payload = MPM2 (615 kg)

Figure 19: Cartesian MP positioning error along axis z_b along the test trajectory

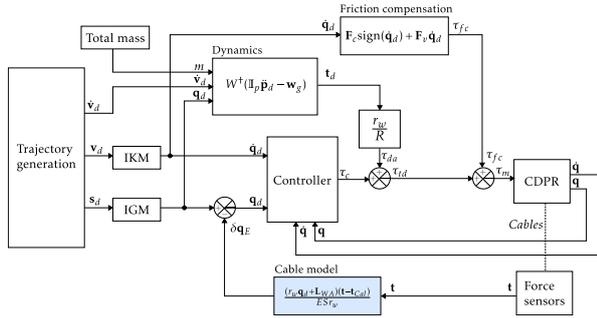


Figure 20: Control scheme with cable elasticity compensation CS4

\mathbf{t} is linked to the cable elongation $\delta \mathbf{l}_E$ by the equation:

$$\mathbf{t} = \frac{ES}{\mathbf{l}} \delta \mathbf{l}_E + \mathbf{t}_{cal} = \frac{ES}{(r_w \mathbf{q}_d + \mathbf{l}_{WA})} r_w \delta \mathbf{q}_E + \mathbf{t}_{cal} \quad (39)$$

with E the cable Young modulus and S its cross-sectional area, \mathbf{l} the nominal cable length, \mathbf{q}_d the nominal motor positions and $\delta \mathbf{q}_E$ the variation in the motor position corresponding to the cable elongation. r_w is the winch radius, \mathbf{l}_{WA} is the vector containing the cable dead lengths from the winches to the pulleys and \mathbf{t}_{cal} is the cable pre-tension measured at the calibration step, when the platform is in a known pose.

Note that the correction term only affects the desired position in the proposed architecture. The cable elasticity compensation should ideally also be applied to the desired speed. However, $\delta \mathbf{q}_E$ is relatively small and is negligible with respect to $\dot{\mathbf{q}}_d$.

6.2. Experimental results

The control scheme CS4 is experimented along the test trajectory for the three loads. The performances of CS4 are compared to those obtained with CS2⁺ without cable elongation compensation.

Figure 21 shows the error along the vertical z -axis between the measured and desired positions of the MP, with CS4-SML and CS2⁺-SML for the three loads. Initially, the cable elongation correction $\delta \mathbf{q}_E$ is not yet applied;

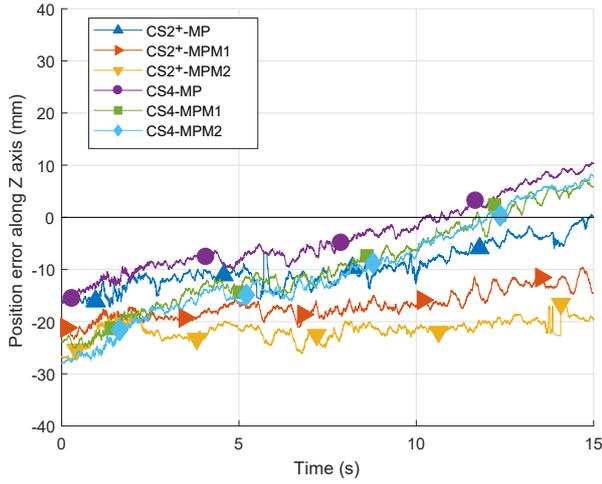
so the MP starts at the same set of motor positions with both CS2⁺-SML and CS4-SML. Since the cable elasticity is still not compensated, the impact of the load on the MP position is clearly visible. The initial error of CS2⁺-SML and CS4-SML is around -15 mm for MP (366 kg), -23 mm for MPM1 (488 kg) and -28 mm for MPM2 (615 kg).

However, once the motion starts the CS4 control scheme applies the motor angular correction $\delta \mathbf{q}_E$ to the desired motor position \mathbf{q}_d , and the errors measured for the three different loads converge. After 15 seconds, the errors measured with the CS4 control scheme are within 4 mm of each other, between 6 and 10 mm, while CS2⁺ leads to a difference of around 20 mm between the MP position with the 366 kg compared to 615 kg.

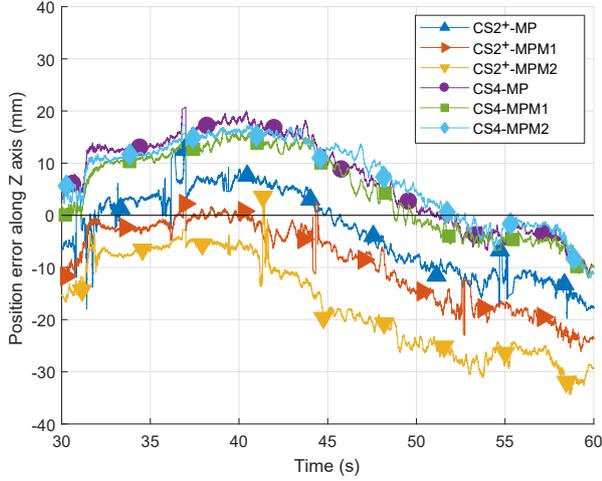
Figure 21(b) is focused on the last 30 seconds of the trajectory. It appears that the MP position repeatability is greatly improved with the CS4 despite the load variation: the curves CS4-MP, CS4-MPM1 and CS4-MPM2 are very close during the entire second half of the trajectory. On the other hand, for the CS2⁺ that does not include the elasticity compensation, the MP ends the trajectory with the same errors than at the beginning, with a spread of 15 mm between the 366 kg and the 615 kg cases. Finally, Figure 22 displays the mean positioning error along the second part of the trajectory for the three loads obtained with CS4 and CS2⁺. The MP positioning error is greatly reduced with the cable elasticity compensation, and should therefore be used for the handling application when applicable. Some modeling errors are still to be corrected in order to reduce errors along the trajectory, but automated trajectories will be more reliable as the MP position is robust to the mass variation.

7. Conclusion

This paper aimed to provide recommendations on the selection of a control strategy depending on the available information on the



(a) First 15 seconds of the test trajectory.



(b) Last 30 seconds of the test trajectory

Figure 21: MP center P initial position error along Z axis (mm) with control schemes $CS2^+$ and $CS4$, for the first 15 seconds and last 30 seconds of the test trajectory.

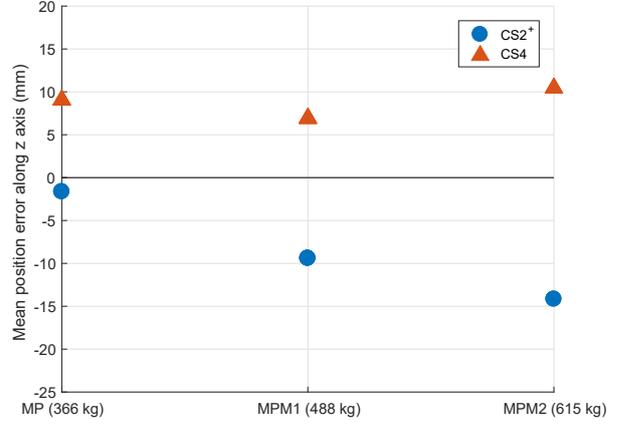


Figure 22: Mean Cartesian position error P along axis z_b (mm) with control schemes $CS2^+$ and $CS4$ along the second half of the trajectory.

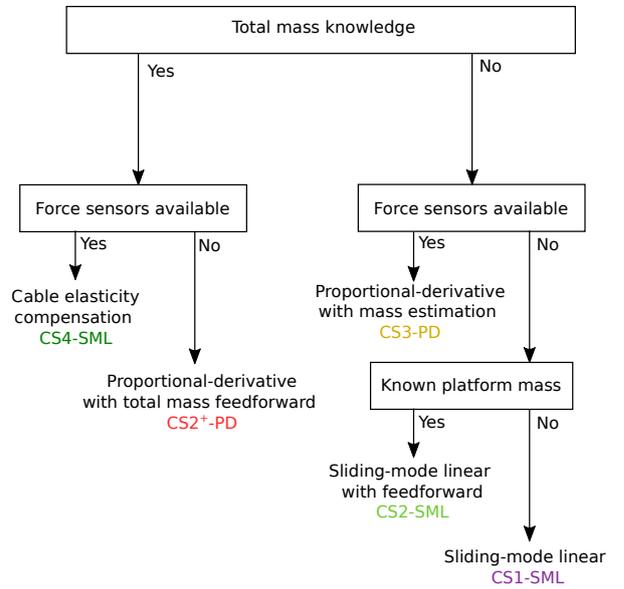


Figure 23: Recommended control schemes based on the task, the available information and sensors.

carried mass, and the presence or absence of force sensors for pick-and-place operations of metal plates with a Cable-Driven Parallel Robot (CDPR).

Three scenarios are defined to compare the performances of adapted control schemes. The control scheme CS1 makes no assumption on the masses of the moving parts. The control scheme CS2 includes a feedforward term for the compensation of the known moving-platform (MP) mass. If the total carried mass is known, including the metal plate, the control scheme is referenced as CS2⁺. The control scheme CS3 takes advantage of force sensors to measure cable tensions, to implement a real-time payload estimation and compensation. Two families of controllers have been experimented in combination with these control schemes: a proportional-derivative (PD) controller, and a recent controller balancing between sliding mode and linear control (SML).

Figure 23 summarizes the main contribution of the paper in the form of a decision tree to suggest the selection of a control architecture between control schemes CS1, CS2 and CS3, and controllers PD and SML depending on the system information and equipment.

In the most favorable conditions, *i.e.* when the total payload is known, the CS2-PD⁺ architecture can be applied for its good accuracy and smooth control signal. This scheme, however, is very restrictive due to the knowledge of the carried mass. If the CDPR is equipped with force sensors, the mass estimation can be implemented with CS3-PD to automatically adapt the mass compensation term.

Nonetheless, if no information is available on the load mass or on the cable tensions, the proposed control schemes implementing the recent SML controller are good options for their stability and robustness towards uncertainties, as both CS1-SML and CS2-SML achieve high accuracy once the controller has reached steady state. It should be noted that the SML controller could easily be further tuned for CDPR

in order to increase the correction speed. The sliding mode based control scheme being the most robust, its application to the most favorable cases should also be considered, with the possibility of implementing higher order control laws to go towards control smoothness.

Finally, a fourth control scheme CS4 is synthesized in the paper by adding a cable elasticity compensation term to the control scheme CS2⁺-SML in order to reduce the MP positioning errors and further improve the overall control robustness towards the mass variations. This addition greatly improves the MP positioning accuracy and repeatability in spite of load changes. As such, this term should ideally be applied to any control scheme. If force sensors are available, the implementation can be realized as described in CS4. A theoretical computation of the elongation based should be experimented in the future for implementation in the case where the CDPR is not equipped with such sensors.

Acknowledgments

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

- [1] Albus, J., Bostelman, R., Dagalakis, N., 1992. The NIST SPIDER, A robot crane. *Journal of Research of the National Institute of Standards and Technology* volume 97, 373. doi:10.6028/jres.097.016.
- [2] Briot, S., Caro, S., Germain, C., 2017. Design procedure for a fast and accurate parallel manipulator. *Journal of Mechanisms and Robotics* 9.
- [3] Edwards, C., Spurgeon, S., 1998. *Sliding mode control: theory and applications*. CRC Press. URL: <https://www.taylorfrancis.com/books/9781498701822>, doi:10.1201/9781498701822.
- [4] El-Ghazaly, G., Gouttefarde, M., Creuze, V., 2015. Adaptive terminal sliding mode control of a redundantly-actuated cable-driven parallel manipulator: CoGiRo, in: *Mechanisms and Machine Science*. Springer, Cham. volume 32,

- pp. 179–200. URL: http://link.springer.com/10.1007/978-3-319-09489-2_13, doi:10.1007/978-3-319-09489-2_13.
- [5] Fortin-Coté, A., Cardou, P., 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), Hong Kong. pp. 819–925.
- [6] Gagliardini, L., 2016. Discrete reconfigurations of cable-driven parallel robots. Phd thesis, centrale nantes.
- [7] Gagliardini, L., Gouttefarde, M., Caro, S., 2018. Determination of a Dynamic Feasible Workspace for Cable-Driven Parallel Robots. *Advances in Robot Kinematics 2016 volume 4*, 361–370. URL: http://link.springer.com/10.1007/978-3-319-56802-7_38, doi:10.1007/978-3-319-56802-7_38.
- [8] Hu, J., Li, C., Chen, Z., Yao, B., 2020. Precision motion control of a 6-dofs industrial robot with accurate payload estimation. *IEEE/ASME Transactions on Mechatronics* 25, 1821–1829.
- [9] Kawamura, S., Kino, H., Won, C., Kamawura, S., Kino, H., Won, C., 2000. High-speed manipulation by using parallel wire-driven robots. *Robotica* 18, 13–21. URL: <http://www.journals.cambridge.org/abstract/S0263574799002477>, doi:10.1017/S0263574799002477.
- [10] Khalil, W., Dombre, E., 2004. Modeling, Identification and Control of Robots. Butterworth-Heinemann. doi:10.1016/B978-1-903996-66-9.X5000-3.
- [11] Kraus, W., Schmidt, V., Rajendra, P., Pott, A., 2013. Load identification and compensation for a Cable-Driven parallel robot, in: Proceedings - IEEE International Conference on Robotics and Automation, pp. 2485–2490. doi:10.1109/ICRA.2013.6630915.
- [12] Kraus, W., Schmidt, V., Rajendra, P., Pott, A., 2014. System identification and cable force control for a cable-driven parallel robot with industrial servo drives, in: Proceedings - IEEE International Conference on Robotics and Automation, pp. 5921–5926. doi:10.1109/ICRA.2014.6907731.
- [13] Lamaury, J., 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), Karlsruhe, Germany. pp. 4659–4664. doi:10.1109/ICRA.2013.6631240.
- [14] Lamaury, J., Gouttefarde, M., Chemori, A., 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), Tokyo, Japan. pp. 4879–4886. doi:10.1109/IROS.2013.6697060.
- [15] Levant, A., 1993. Sliding order and sliding accuracy in sliding mode control. *International Journal of Control* 58, 1247–1263. doi:10.1080/00207179308923053.
- [16] Merlet, J.P., 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), pp. 2834–2839. doi:10.1109/ROBOT.2010.5509299.
- [17] Picard, É., Caro, S., Claveau, F., Plestan, F., 2018a. 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), Madrid, Spain. pp. 1429–1436. doi:10.1109/IROS.2018.8594171.
- [18] Picard, E., Caro, S., Plestan, F., Claveau, F., 2018b. Control solution for a cable driven parallel robot with highly variable payload, in: Proceedings of the 2018 ASME Design Engineering Technical Conference, Quebec City, QC. doi:10.1115/DETC2018-85304.
- [19] Santos, J.C., Chemori, A., Gouttefarde, M., 2019. Model Predictive Control of Large-Dimension Cable-Driven Parallel Robots, Springer, Cham, pp. 221–232. URL: http://link.springer.com/10.1007/978-3-030-20751-9_19, doi:10.1007/978-3-030-20751-9_19.
- [20] Schenk, C., Masone, C., Pott, A., Bühlhoff, H.H., 2018. Application of a differentiator-based adaptive super-twisting controller for a redundant cable-driven parallel robot, in: Mechanisms and Machine Science, pp. 254–267. doi:10.1007/978-3-319-61431-1_22.
- [21] Shtessel, Y., Edwards, C., Fridman, L., Levant, A., 2014. Sliding mode control and observation. Control Engineering, Springer New York, New York, NY. URL: <http://link.springer.com/10.1007/978-0-8176-4893-0>, doi:10.1007/978-0-8176-4893-0.
- [22] Tahoumi, E., Ghanes, M., Plestan, F., Barbot, J.P., 2018a. A New Controller Switching between Linear and Twisting Algorithms. Proceedings of the 2018 American Control Conference , 6150–6155doi:10.23919/ACC.2018.8431897.
- [23] Tahoumi, E., Plestan, F., Ghanes, M., 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), pp. 2368–2373. doi:10.23919/ECC.2018.8550181.
- [24] Utkin, V.I., 1992. Sliding Modes in Control and Optimization. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-84379-2.

- [25] Vafaei, A., Aref, M.M., 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), pp. 650–655. doi:10.1109/ROBOT.2010.5509991.
- [26] Yao, R., Tang, X., Wang, J., Huang, P., 2010. Dimensional optimization design of the four-cable-driven parallel manipulator in fast. IEEE/ASME Transactions on Mechatronics 15, 932–941. doi:10.1109/TMECH.2009.2035922.
- [27] Zeinali, M., 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, pp. 319–327. doi:10.1115/DETC2010-29153.
- [28] Ziegler, J.G., Nichols, N.B., 1995. Optimum settings for automatic controllers. InTech volume 42, 94–100. doi:10.1115/1.2899060.