Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 550–555

# Model-free cable robot control [★][★★]

**Franco Blanchini** [*] **Luca Della Schiava** [****] **Gianfranco Fenu** [**]
**Giulia Giordano** [***] **Felice Andrea Pellegrino** [**] **Erica Salvato** [**]

[*] *Dipartimento di Scienze Matematiche, Informatiche e Fisiche, Università degli Studi di Udine, 33100 Udine, Italy.* blanchini@uniud.it
[**] *Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste, 34127 Trieste, Italy.* {fenu,fapellegrino}@units.it, erica.salvato@dia.units.it
[***] *Dipartimento di Ingegneria Industriale, Università degli Studi di Trento, 38123 Povo, Italy.* giulia.giordano@unitn.it
[****] *Electrolux Italia S.p.A., Porcia, 33080, Italy.* luca.della-schiava@electrolux.com

**Abstract:** This paper proposes a technique to control a cable robot in the total absence of a model and its parameters. The cable robot is actuated by three motors whose data, including exact positions, pulley diameters, and nominal cable length, are unknown. We just assume to have a very rough knowledge of lower and upper bounds for the partial derivatives of the relation between the cable lengths and the end-effector space coordinates. A structured-light sensor measures the end-effector position, and the goal is to drive it to a designated point. An algorithm is proposed with guaranteed convergence based on the so-called model-free plant tuning approach. No learning stage is required. Experimental results are reported.

*Keywords:* Plant-tuning, Model-free, Cable Robot, Min-max

## 1. INTRODUCTION

*Model-free plant tuning* aims to determine the input vector of a static plant, governed by a smooth function, so that the output vector assumes prescribed values, in the absence of a model. The only required information is the sign of the partial derivatives of the unknown input-output function, along with (even rough) upper and lower bounds. More in general, it is sufficient to know that the Jacobian of the unknown function is confined to a given polytope.

Tuning a plant whose model is not known exactly often requires a frustrating trial-and-error approach, and, for example, when attempting to set an output to the desired value, the unknown interactions among the variables can unpredictably drive the other outputs out of tune.

Under the assumption that the Jacobian is confined in a polytope (or, more in general, in a convex and compact set) and that it is robustly non-singular, a tuning scheme with guaranteed convergence has been proposed in our previous work (Blanchini et al. 2015, 2017). The results technically rely on the min-max theorem (Luenberger 1969) and on a suitable Lyapunov-like function.

Similar approaches have been previously proposed in the literature for robust stabilisation (Gutman & Leitmann 1976; Gutman 1979; Meılakhs 1979; Blanchini 2000; Blanchini & Pesenti 2001). However, in the model-free plant-tuning case there is nothing to be stabilised and the Lyapunov-like function is not defined in the state-space. Possible analogies with methods for parameter tuning (Åström 1983; Fradkov 1980), iterative learning control (Ahn et al. 2007; Bristow et al. 2006), multi-dimensional extremum-seeking techniques (Tan et al. 2006; Khong et al. 2013; Nešić et al. 2013) and robust optimisation (Beyer & Sendhoff 2007) are thoroughly discussed by Blanchini et al. (2017). The case of plants with parasitic dynamics has been considered by Blanchini et al. 2017b. A discrete-time iteration technique has been proposed by Blanchini et al. (2017a).

In this paper, we consider the specific problem of controlling a cable robot (see Fig. 1) whose model is unknown. There is no information about the exact position of the actuators, the nominal cable length, and the pulley diameters. We only require that a suitably placed vision sensor detects the current position of the end-effector. Based on a rough knowledge of the Jacobian entry bounds, we prove that our model-free plant tuning scheme ensures exact convergence to the target. The essential advantage is that no learning phase is necessary and the convergence is ensured at the first shot.

We provide experimental results. Accompanying videos are available online.

## 2. PROBLEM STATEMENT

The terms of our problem are the following.

[★] This work has been partially supported by the Italian Ministry for Research in the framework of the 2017 Program for Research Projects of National Interest (PRIN), Grant no. 2017YKXYXJ.
[★★]This study was carried out within the PNRR research activities of the consortium iNEST (Interconnected North-Est Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.5 – D.D. 1058 23/06/2022, ECS_00000043). This manuscript reflects only the Authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

10.1016/j.ifacol.2023.10.1625

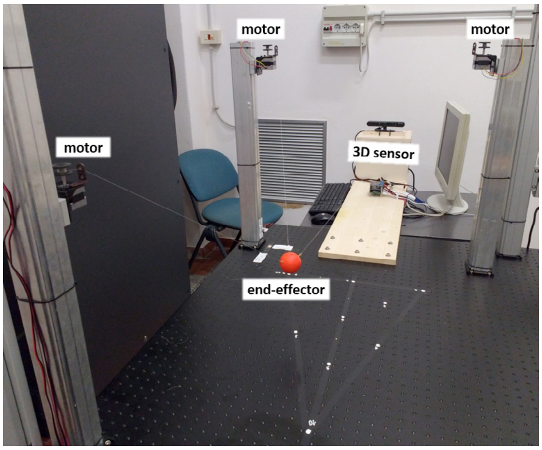

Fig. 1. The experimental setup consists of a 3D sensor (specifically, an RGB-D camera) and a cable robot, having three motors and a red ball end-effector.

- The end-effector is connected via cables to three motors located in $A$, $B$, and $C$, whose angular speed can be controlled.
- The pulley diameters are not known.
- The position of each motor is unknown, but we assume that they are all at the same level with respect to the ground.
- The reference frame has the $z$ axis orthogonal to the ground, with positive direction toward the motors, while the $x$ and $y$ axes are parallel to the ground (as in Figure 2).
- A vision system reveals the coordinates $x$, $y$, and $z$ of the end-effector (i.e. the center of the red ball in Figure 2) in the reference frame.
- The vision system allows to measure the positioning error $x-x_r$, $y-y_r$, $z-z_r$, where $x_r$, $y_r$, and $z_r$ are the coordinates of a reference point.
- The $y$ axis is parallel to the line connecting the position of motors $A$ and $B$, with positive direction from $B$ to $A$.
- The triangle drawn by the motors is acute, i.e., $y_B < y_C < y_A$.
- We can control the motors' speed, proportional to the cable length derivatives.
- The end-effector lies below the level of the motors, and its projection is located within a bounded region inside the triangle resulting from the projection of the motors' position on the ground.

The configuration is represented in Figure 2. The goal is to steer the robot end-effector to a reference point, in a model-free fashion.

## 3. PRELIMINARY THEORETICAL RESULTS

Given an unknown function that relates the output $y$ to the input $u$, we aim at driving the output to a desired value (which we can set to zero without loss of generality) by means of a suitable input sequence $u$. A crucial assumption is that the updating of $u$ is performed at discrete time instants: $u_k$.

*Problem 1.* Given the static plant

$$e = g(u), \tag{1}$$

where $e \in \mathbb{R}^p$ is the error, $g : \mathbb{R}^m \to \mathbb{R}^p$, $p \le m$, is a continuously differentiable function, and $g(\bar{u}) = 0$ for some unique *unknown* $\bar{u}$, find a dynamic algorithm such that, as $t \to \infty$:

$$e(t) \to 0, \tag{2}$$

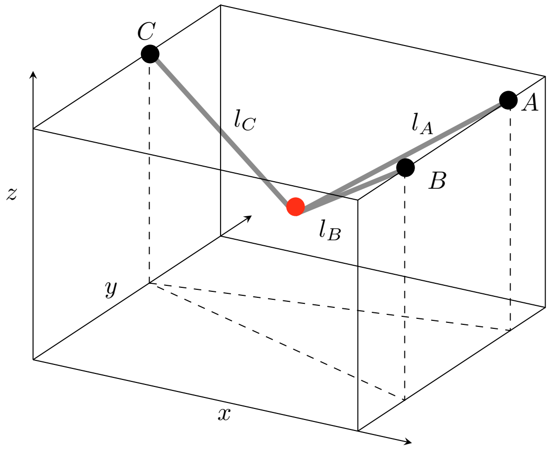

Fig. 2. Setup scheme: the motors are located in $A$, $B$ and $C$; $l_A$ is the distance from the motor located in $A$ to the end-effector. Similarly, $l_B$ and $l_C$ are, respectively, the distances from the motors located in $B$ and $C$ to the end-effector.

$$u(t) \to \bar{u}, \tag{3}$$

where $\bar{u}$ solves the equation

$$0 = g(u). \tag{4}$$

◇

*Assumption 2.* The following inclusion holds:

$$G_u \doteq \left[\frac{\partial g}{\partial u}\right] \in \mathscr{G} \tag{5}$$

where $\mathscr{G}$ is a known polytope of matrices, with vertices $G_i$:

$$\mathscr{G} = \left\{ G = \sum_{i=1}^{r} G_i \alpha_i, \;\; \alpha_i \ge 0, \;\; \sum_{i=1}^{r} \alpha_i = 1 \right\}. \tag{6}$$

*Theorem 1.* (Blanchini et al. (2017)) Assume that the polytope $\mathscr{G}$ is nonsingular. Then, there exists a function $\Phi(e)$ such that the control scheme

$$\dot{u} = \Phi(e)$$

drives $e$ to 0 (the reference point). The function can be chosen as

$$\Phi(e) = -\gamma v^*, \tag{7}$$

where $\gamma > 0$ regulates the convergence speed, and

$$v^* = \arg\min_{M \in \mathscr{G}} \|y^\top M\|. \tag{8}$$

In general, to apply the control, namely, to compute $\Phi$, we need to solve the optimization problem (8), online. This reduces to linear quadratic optimization.

## 4. PROBLEM SOLUTION

To find the solution, as a first step, we need to check that the following *unknown* function, which gives the position error $e = [x-x_r, \; y-y_r, \; z-z_r]^\top$ as a function of the cable lengths $u = [l_A, \; l_B, \; l_C]^\top$, has a Jacobian that can be included in a robustly non-singular polytope:

$$x - x_r = g_x(l_A, l_B, l_C) \tag{9}$$

$$y - y_r = g_y(l_A, l_B, l_C) \tag{10}$$

$$z - z_r = g_z(l_A, l_B, l_C). \tag{11}$$

It is important to note that the Jacobian has the following sign pattern matrix [1]

$$\Sigma = \begin{bmatrix} - & - & + \\ - & + & 0 \\ - & - & - \end{bmatrix}, \tag{12}$$

meaning that $\text{sign}(J_{ik}) = \Sigma_{ik}$. No sophisticated analysis is needed (but we provide it later anyhow) to see that $z$ is a decreasing function of $l_A$, $l_B$, and $l_C$, which explains the $-$ signs on the last row. Any variation of $l_C$ does not change $y$, which explains $\Sigma_{23} = 0$. Increasing $l_A$ decreases $y$, and increasing $l_B$ increases $y$, which explains the $-$ and $+$ signs on the second row, respectively. Finally, observing the first row of $\Sigma$, $x$ is negatively affected by an increase of $l_A$ and $l_B$, and positively affected by an increase of $l_C$.

Then, assuming to have upper and lower bounds on the derivatives:

$$0 < \alpha \leq |J_{ik}| \leq \beta \tag{13}$$

we see that the Jacobian belongs to an interval family

$$J \in \left\{ \begin{bmatrix} [-\beta,-\alpha] & [-\beta,-\alpha] & [+\alpha,+\beta] \\ [-\beta,-\alpha] & [+\alpha,+\beta] & 0 \\ [-\beta,-\alpha] & [-\beta,-\alpha] & [-\beta,-\alpha] \end{bmatrix} \right\}. \tag{14}$$

*Proposition 1.* The interval family (14) is robustly non–singular.

**Proof.** It is an immediate consequence of the fact the determinant

$$\det(J) = J_{11}J_{22}J_{33} + J_{13}J_{21}J_{32} - J_{31}J_{22}J_{13} - J_{33}J_{21}J_{12}$$

is strictly positive, as can be seen by taking into account the sign of the entries $J_{ik}$. By continuity, $\det(J)$ has a positive minimum. □

To derive the extrema $\alpha$ and $\beta$, we first notice that the result does not depend on their magnitude, which, consequently, can be set very roughly.

In addition, it is also important to note that finding the exact values for $\alpha$ and $\beta$ is not crucial, while the $\beta/\alpha$ ratio of the adopted bounds must be greater than that of the "true" $\hat{\alpha}$ and $\hat{\beta}$ bounds.

Assume that the "true" positive bounds are $\hat{\alpha}$ and $\hat{\beta}$ (unknown), and that $\alpha$ and $\beta$ are the problem bounds imposed in (14). Then, the proposed approach works, as long as:

$$\frac{\hat{\beta}}{\hat{\alpha}} \leq \frac{\beta}{\alpha}. \tag{15}$$

This property follows from the fact that $\gamma > 0$ is arbitrary. Indeed, if the derivatives are not in the adopted bounds (13), then if (15) holds, and

$$\hat{\alpha} \leq |J_{ik}| \leq \hat{\beta},$$

for some *unknown* $\sigma > 0$, we have also:

$$\alpha \leq |\sigma J_{ik}| = |\hat{J}_{ik}| \leq \beta,$$

leading to:

$$\dot{e} = \frac{1}{\sigma}\left[\sigma \frac{\partial g}{\partial u}\right]\dot{u} = \hat{J}\frac{-\gamma v}{\sigma} = \hat{J}\hat{v}.$$

Since the entries of $\hat{J}$ are in the interval $[\hat{\alpha}, \hat{\beta}]$, we just need to consider the scaled input $\hat{v}$ and apply the theory. Actually, we apply $v$, not $\hat{v}$, but, since $\gamma$ is arbitrary, the input can be scaled without affecting the final result.

*Proposition 2.* For any adopted bounds $[\alpha, \beta]$, the proposed approach converges provided that the true bounds $[\hat{\alpha}, \hat{\beta}]$, satisfy (15).

*Remark 1.* Note that $\Sigma_{23} = 0$ is crucial, and it is ensured by the assumption that the $y$ axis is parallel to the line connecting the positions of motors $A$ and $B$, and that the $z$-coordinates of these two motors are the same (i.e., $x_A = x_B$, and $z_A = z_B$).

Although bounds $[\alpha, \beta]$ on the derivative can be practically found, later on, we will provide some additional considerations on the Jacobian, useful to derive these bounds.

Finally, we have to discuss the fact that the pulleys have unknown diameters, hence we cannot govern directly the inputs $(l_A, l_B, l_C)$. Precisely, we have assumed:

$$\frac{d}{dt}(l_A, l_B, l_C) = (v_1, v_2, v_3).$$

If the true control variables are the motors' speeds, clearly we have that:

$$\frac{d}{dt}(l_A, l_B, l_C) = r(\omega_A, \omega_B, \omega_C),$$

where $r$ is the unknown pulleys radius, while $\omega_A$, $\omega_B$, and $\omega_C$ are the angular speeds of the motors $A$, $B$, and $C$, respectively. But, again, by considering a scaled input $(\hat{v}_1, \hat{v}_2, \hat{v}_3) = r(\omega_A, \omega_B, \omega_C)$, with unknown $r > 0$, the final result is not affected by this choice, and the approach applies.

### *4.1 An explicit formula for control*

We have seen that, in order to implement the control, we need to solve, online, the minimum Euclidean norm problem:

$$\min_{M \in \mathscr{M}} \|y^\top M\|.$$

Since $M$ is an interval matrix, the above problem is decoupled into three sub-problems, one for each component of $v$:

$$v_j = y_1 m_{1j} + y_2 m_{2j} + y_3 m_{3j},$$

where:

$$v^2 = \min_{m_{1j}, m_{2j}, m_{3j}} (y_1 m_{1j} + y_2 m_{2j} + y_3 m_{3j})^2, \quad j = 1, 2, 3.$$

Let us consider the equivalent problem of minimizing $|\cdot|$ instead of $(\cdot)^2$:

$$v = y_1 z_1 + y_2 z_2 + y_3 z_3, \quad |v| = \min_{z_i^- \leq z_i \leq z_i^+} |y_1 z_1 + y_2 z_2 + y_3 z_3|.$$

The resulting domain is a box.

Consider, now, the following two linear programming problems:

$$\eta = \min_{z_i^- \leq z_i \leq z_i^+} \; y_1 z_1 + y_2 z_2 + y_3 z_3$$
$$\mu = \max_{z_i^- \leq z_i \leq z_i^+} \; y_1 z_1 + y_2 z_2 + y_3 z_3,$$

and note that $\eta < \mu$. Then:

$$v = \begin{cases} \eta & \text{if } 0 < \eta < \mu \\ 0 & \text{if } \eta \leq 0 \leq \mu \\ \mu & \text{if } \eta < \mu < 0 \end{cases}.$$

For the proof, we just need to observe that if the two linear programming problems have opposite signs at the optimum, then there is a choice of $z$ for which $y_1 z_1 + y_2 z_2 + y_3 z_3 = 0$. Otherwise, if they have the same sign, we take the minimum in absolute value.

[1] see Van den Driessche et al. (2018) for a formal definition.

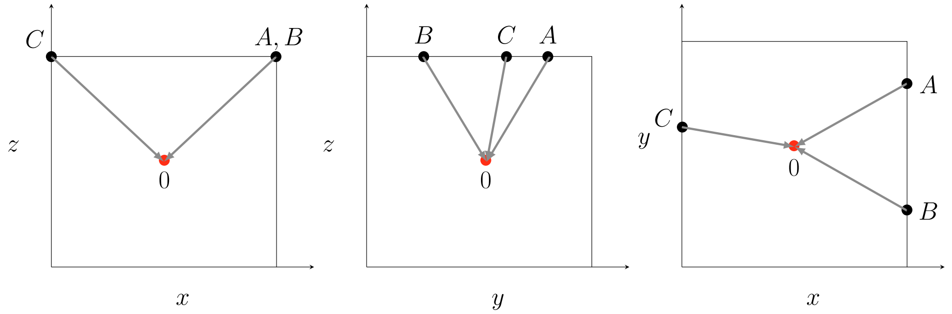

Fig. 3. The projection of the cable robot in Fig. 2 on the $z-x$, $z-y$, and $x-y$ planes.

The computation of $\eta$ and $\mu$ is trivial. Consider the function:

$$\sigma_i(y_i) = \begin{cases} z_i^+ & \text{if } \ y_i > 0 \\ z_i^- & \text{if } \ y_i < 0 \end{cases},$$

where $\sigma_i(y_i)$ is inside $[z_i^-, z_i^+]$ if $y_i = 0$. Then:

$$\mu = y_1\sigma_1(y_1) + y_2\sigma_2(y_2) + y_3\sigma_3(y_3)$$

and

$$\eta = y_1\sigma_1(-y_1) + y_2\sigma_2(-y_2) + y_3\sigma_3(-y_3).$$

## 5. ANALYSIS OF THE JACOBIAN AND ITS BOUNDS

As mentioned before, we analyse the Jacobian to prove the heuristic considerations in Section 4.

Unfortunately, we do not have $g$ explicitly, so we resort to the inverse function $h \doteq g^{-1}$, compute its Jacobian, and find the inverse, since:

$$\frac{\partial g}{\partial u} = \left[\frac{\partial h}{\partial e}\right]^{-1}.$$

We can write (9)–(11) as:

$$x = g_x(l_A, l_B, l_C) + x_r \tag{16}$$
$$y = g_y(l_A, l_B, l_C) + y_r \tag{17}$$
$$z = g_z(l_A, l_B, l_C) + z_r, \tag{18}$$

and note that *the Jacobian does not change*, because $x_r$, $y_r$, and $z_r$ are constant reference values. The inverse function is:

$$l_A = h_A(x,y,z) = \sqrt{(x-x_A)^2 + (y-y_A)^2 + (z-z_A)^2} \tag{19}$$
$$l_B = h_B(x,y,z) = \sqrt{(x-x_B)^2 + (y-y_B)^2 + (z-z_B)^2} \tag{20}$$
$$l_C = h_C(x,y,z) = \sqrt{(x-x_C)^2 + (y-y_C)^2 + (z-z_C)^2}, \tag{21}$$

having an inverse Jacobian:

$$J^{-1} = \begin{bmatrix} l_A & 0 & 0 \\ 0 & l_B & 0 \\ 0 & 0 & l_C \end{bmatrix}^{-1} \begin{bmatrix} (x-x_A) & (y-y_A) & (z-z_A) \\ (x-x_B) & (y-y_B) & (z-z_B) \\ (x-x_C) & (y-y_C) & (z-z_C) \end{bmatrix}.$$

Now we remind that

$$(z-z_A) = (z-z_B) = (z-z_C), \quad \text{and} \quad (x-x_A) = (x-x_B).$$

As a consequence, we have:

$$\det[J^{-1}] = (l_A l_B l_C)^{-1}(y_A - y_B)(x_A - x_C)(z_A - z),$$

which is strictly positive as expected. Consequently, the Jacobian matrix takes the form:

$$J = \frac{1}{(y_A-y_B)(x_A-x_C)(z_A-z)} \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & 0 \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \begin{bmatrix} l_A & 0 & 0 \\ 0 & l_B & 0 \\ 0 & 0 & l_C \end{bmatrix}.$$

Applying elementary algebra, we see that the term $\phi_{ik}$ is the signed magnitude of the cross-product among vectors, which are projections on the principal 2-dimensional spaces (the mentioned projections are depicted in Fig. 3).

More precisely, we have:

$$\phi_{11} = (-1)^{1+1}\left(\pm \left\| \begin{bmatrix} 0 \\ (y-y_B) \\ (z-z_B) \end{bmatrix} \times \begin{bmatrix} 0 \\ (y-y_C) \\ (z-z_C) \end{bmatrix} \right\| \right),$$

$$\phi_{21} = (-1)^{2+1}\left(\pm \left\| \begin{bmatrix} (x-x_B) \\ 0 \\ (z-z_B) \end{bmatrix} \times \begin{bmatrix} (x-x_C) \\ 0 \\ (z-z_C) \end{bmatrix} \right\| \right)$$

and so on, where the sign depends on the angle between the vectors.

Examining these vector products, the signs in (12) can be verified. In particular, since $x - x_A = x - x_B$ and $z - z_A = z - z_B$, we have

$$\phi_{23} = 0,$$

which ensures the structural non-singularity.

### 5.1 Bounds for the Jacobian entries

To provide bounds for the entries of $J$ we remind that the signed magnitude of the cross product is given by $\pm\|v_k\|\|v_h\|\sin(\varphi_{hk})$, hence the magnitude of $\phi_{hk}$ is:

$$|\phi_{hk}| \leq \|v_k\|\|v_h\||\sin(\varphi_{hk})|,$$

where $\varphi_{hk}$ is the angle formed by the vectors. The generic entry of the Jacobian is

$$J_{hk} = \frac{\ell}{(y_A-y_B)(x_A-x_C)(z_A-z)}\phi_{hk}$$

where $\ell$ equals $l_A$ for $k=1$, $l_B$ for $k=2$, and $l_C$ for $k=3$, hence

$$|J_{hk}| = \frac{\ell}{(y_A-y_B)(x_A-x_C)(z_A-z)}\|v_k\|\|v_h\|\,|\sin(\varphi_{hk})|\,.$$

We can reasonably assume bounds:

$$0 < \sigma_{min} \leq \sin(\varphi_{hk}) \leq 1.$$

The length of the projected vectors can be bounded as follows:

$$(z_A - z_{max}) \leq \|v_k\| \leq l_{max},$$

where $z_{max}$ is the maximum level of the end-effector, and $l_{max}$ is the maximum cable length (an overbound). A bound for the determinant can be derived by (rough) bounds on the geometry of the work-space:

$$\begin{aligned}(y_A-y_B)_{min}\cdot(x_A-x_C)_{min}\cdot(z_A-z_{max}) &\leq \det[J^{-1}] \\ &\leq (y_A-y_B)_{max}\cdot(x_A-x_C)_{max}\cdot\, l_{max}.\end{aligned} \tag{22}$$

The adjoint terms, with the exception of $\phi_{23} = 0$, are all bounded as:

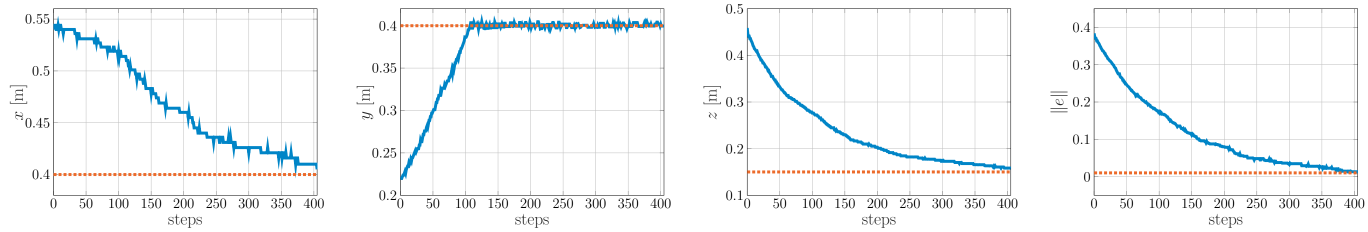

Fig. 4. End-effector trajectory (coordinate by coordinate), from $P_0$ to $P_T$, under the proposed control law. Starting from the left: the step-by-step behaviour of the end-effector $x$-coordinate, $y$-coordinate, $z$-coordinate, and finally of the error norm $||e||$ (blue lines). The dotted orange lines represent the coordinate targets and the error norm threshold.

$$(z_A - z_{max})^2 \sigma_{min} \leq \phi_{ik} \leq l_{max}^2, \qquad (23)$$

and hence, along with (22), we can determine the lower and upper bound for the entries of $J$.

In the experimental setup (Fig. 1) we take: $(y_A - y_B)_{min} = 0.5m$, $(y_A - y_B)_{max} = 1.5m$, $(x_A - x_C)_{min} = 0.5m$, $(x_A - x_C)_{max} = 1.5m$, $(z_A - z_{max} = 0.10m$, $(z_A - z)_{max} = 1.0m$, $\sigma_{min} = 0.1$.

*Remark 2.* It is immediate to see that $\sin(\varphi_{hk}) = 0$ is a singular configuration to be avoided.

## 6. LABORATORY RESULTS

As an application of the proposed technique, let us consider the experimental setup in Fig. 1.

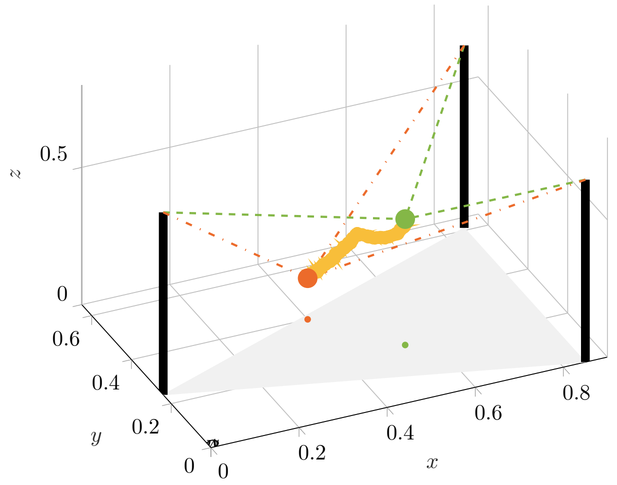

Fig. 5. 3D view of the experimental result: the end-effector initial and final positions are represented by the green and orange spheres, respectively, while the yellow cylindrical trajectory indicates the different positions assumed by the end-effector during the experiment. The initial and final configuration of the robot cables are represented as green and orange dashed lines, respectively. Finally, the small green and orange dots on the $x-y$ plane indicate the projections on the plane of the initial and final position of the sphere, respectively.

It consists of:

- three stepper motors, with 200 steps/ rev and 12 V rated voltage, connected with nylon cables to a red ball acting as the end-effector of the resulting cable robot;
- a Raspberry Pi 3 B, running Raspbian OS (a Linux OS distribution, based on Debian and optimized for the Raspberry Pi), and equipped with two Adafruit Motor HAT boards, used to drive the motors;
- a 3D sensor PrimeSense Carmine v. 1.09, handled by the Raspberry Pi using OpenNI and PrimeSense Sensor libraries (respectively v. 1.5.7 and v. 5.1.6.6), which captures real-time images of the red ball in the 3D space.

Each motor is attached to a 1.20 m-long ThorLabs Construction rail [2] (see Fig. 1), and the whole structure has been assembled on a Breadboard ThorLabs table [3] .

The rails (three in total) are located in accordance with the scheme reported in Fig. 2. In particular, in compliance with the reference frame depicted in Fig. 2, and the requirements $x_A = x_B$, $z_A = z_B = z_C$, and $y_B < y_C < y_A$, the motors are located at:

$$A : (0.850, 0.610, 0.665) \text{ m}$$
$$B : (0.850, 0.000, 0.665) \text{ m}$$
$$C : (0.000, 0.240, 0.665) \text{ m}.$$

We denote by *working area* the triangular region obtained by projecting the motor pillars onto the $x-y$ plane.

The vision sensor is instead located at $(1.737, 0.286, 0.225)$ m, with respect to the same reference frame.

The Raspberry Pi runs in real-time two different algorithms: (i) the one acting as the model-free plant tuning controller, and (ii) the one used to extract the end-effector position in the 3D-space. The latter is pretty straightforward: first, the ball is detected in the RGB image by applying a Difference of Gaussian filter of appropriate size to the red channel only, and searching for the local maxima in the filtered image. Then, the depth image is read in correspondence of the detected position in the filtered image, to obtain the 3D coordinates of the ball. We have performed different experiments whose results are available at `https://youtube.com/playlist?list=PLFFRkiZTiAiTAW49lTc7VGOAsUZZfJhCz`. Here we report only one of them.

For all the performed experiments we set the following lower and upper bounds on the derivatives defined in Eq. 13: $\alpha = 5.0 \cdot 10^{-3}$, $\beta = 5.0 \cdot 10^{-1}$.

The experiment consists of a positioning end-effector task in which we want to bring the red ball centre of the cable robot in Fig. 1 from the initial position [4] $P_0 = (0.540, 0.220, 0.459)$ m, to the target position $P_T = (0.400, 0.400, 0.150)$ m. The algo-

[2] `https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=194` .

[3] `https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=7091` .

[4] all the coordinates are expressed with respect to the above-defined reference frame.

rithm is supposed to reach the target when the position error norm is below a selected threshold: 10 mm.

Notice that the starting point $P_0$ of the end-effector trajectory has been chosen such that its projection on the $x-y$ lies in the working zone of the proposed setup (the green dot in Fig. 5). The target $P_T$ (the orange dot in Fig. 5) was instead placed on an edge of the same area.

In Figures 5 and 4, we report the 3D and 2D trajectories, respectively, obtained by performing the proposed model-free plant tuning control approach.

As already mentioned in the previous sections, the controller ignores the exact position of the actuators, the nominal cable lengths, and the pulley diameters, and can only rely on the knowledge of an approximation of $\alpha$ and $\beta$ for each Jacobian entry, and of the estimated position of the end-effector resulting from image processing of the camera output.

In detail, in Fig. 5 the green ball represents the initial position of the cable robot end-effector, while the orange ball is the target. The yellow cylinder represents the ball trajectory performed by the proposed controller. In Fig. 4, instead, the first three subfigures show the trajectory of the end-effector centre along each axis of the reference fame, under the proposed control law, and with respect to the target coordinate (orange dashed line). In the last subfigure of Fig. 4 we report the norm of the position error.

Analysing the reported results, in particular observing Fig. 4, we can notice that initially (during the first 100 steps of the algorithm) the $y$-coordinate of the end-effector position is brought quickly to the target value and then kept close to it, whereas the $x$ and the $z$ coordinates of the end-effector move more slowly toward their respective target values. This particular choice of the proposed controller led consequently to the "elbow" in the yellow trajectory of Fig. 5.

## 7. CONCLUDING DISCUSSION

We proposed a model-free plant tuning approach for the control of a properly designed cable robot. We performed several experiments on a real platform, highlighting the effectiveness of our solution. The strengths of the proposed controller are twofold: it does not require any robot structural details (i.e., no information is needed on the exact position of the actuators, nominal cable length, and pulley diameters), it requires only a well-placed vision sensor to detect the position of the end-effector, and an approximate knowledge of the Jacobian entry bounds. Under proper non-singularity assumptions, we prove that the presented model-free plant tuning leads to the target achievement. This paper is an example of how the technique proposed in Blanchini et al. (2015, 2017) is able to solve different control problems involving different types of dynamical systems. Investigations to validate the proposed technique in addressing visual servoing problems in robotics are already underway.

## REFERENCES

H.-S. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization", *IEEE Trans. on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 37, pp. 1099-1121, 2007.

K. J. Åström, "Theory and applications of adaptive control — A survey", *Automatica*, vol. 19, no. 5, pp. 471–486, 1983.

H. G. Beyer and B. Sendhoff, "Robust optimization — A comprehensive survey", *Comput. Methods in Appl. Mech. Eng.*, vol. 196, no. 33–34, pp. 3190–3218, 2007.

F. Blanchini, "The gain scheduling and the robust state feedback stabilization problems", *IEEE Trans. Autom. Control*, vol. 45, no. 11, pp. 2061–2070, 2000.

F. Blanchini, G. Fenu, G. Giordano, F. A. Pellegrino, "Plant tuning: a robust Lyapunov approach", *Proc. 54^th^ IEEE Conf. on Decision and Control*, Osaka, December 15-18, 2015.

F. Blanchini, G. Fenu, G. Giordano, F. A. Pellegrino, "Model-free plant tuning", *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2623-2634, June 2017.

F. Blanchini, G. Fenu, G. Giordano, F. A. Pellegrino, "Model-free tuning of plants with parasitic dynamics", *Proc. 57^th^ IEEE Conf. on Decision and Control*, Melbourne, 2017.

F. Blanchini, G. Fenu, G. Giordano, F. A. Pellegrino, Discrete-time trials for tuning without a model, IFAC-PapersOnLine, Volume 50, Issue 1, July 2017, Pages 1539-1544.

F. Blanchini and R. Pesenti, "Min-max control of uncertain multi-inventory systems with multiplicative uncertainties", *IEEE Trans. Autom. Control*, vol. 46, no. 6, pp. 955–959, 2001.

D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, June 2006.

A. Fradkov, "A scheme of speed gradient and its application in problems of adaptive control", *Autom. and Rem. Contr.*, vol. 40, no. 9, pp. 1333–1342, 1980.

M. Grant and S. Boyd. *CVX: Matlab Software for Disciplined Convex Programming, v. 2.1*, http://cvxr.com/cvx, 2014.

S. Gutman, "Uncertain dynamical systems – a Lyapunov min-max approach", *IEEE Trans. Autom. Control*, vol. 24, no. 3, pp. 437–443, 1979.

S. Gutman and G. Leitmann, "Stabilizing control for linear systems with bounded parameter and input uncertainty", Optimization Techniques Modeling and Optimization in the Service of Man Part 2, Lecture Notes in Computer Science, pp. 729-755, 1976.

S. Z. Khong, D. Nešić, Y. Tan, C. Manzie, "Unified frameworks for sampled-data extremum seeking control: global optimisation and multi-unit systems", *Automatica*, vol. 49, no. 9, pp. 2720–2733, 2013.

D. G. Luenberger, *Optimization by vector space methods*. John Wiley & Sons Inc., New York, 1969.

A. M. Meılakhs, "Design of stable control systems subject to parametric perturbation", *Autom. Rem. Control*, vol. 39, no. 10, pp. 1409–1418, 1979.

D. Nešić, A. Mohammadi, C. Manzie, "A framework for extremum seeking control of systems with parameter uncertainties", *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 435–448, 2013.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, USA, 1970.

Y. Tan, D. Nešić and I. Mareels, "On non-local stability properties of extremum seeking control", *Automatica*, vol. 42, no. 6, pp. 889–903, 2006.

P. Van den Driessche, "Sign Pattern Matrices", *Encinas, A., Mitjana, M. (eds) Combinatorial Matrix Theory. Advanced Courses in Mathematics CRM Barcelona. Birkhäuser, Cham*, 2018. `https://doi.org/10.1007/978-3-319-70953-6_2`.