Robotics and Autonomous Systems 193 (2025) 105045

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Position-based visual servo control without hand-eye calibration ☆

Erica Salvato [a,*], Franco Blanchini [b], Gianfranco Fenu [a], Giulia Giordano [c], Felice Andrea Pellegrino [a]

[a] Department of Engineering and Architecture, University of Trieste, Via Alfonso Valerio 6/1, Trieste, 34127, Italy
[b] Dipartimento di Scienze Matematiche, Informatiche e Fisiche, Università degli Studi di Udine, Via delle Scienze 206, Udine, 33100, Italy
[c] Department of Industrial Engineering, University of Trento, Via Sommarive, 9, Trento, 38123, Italy

ARTICLE INFO

Keywords:
Visual servoing
Robot manipulators
Lyapunov control
Robust control

ABSTRACT

We deal with the position-based visual servoing control of a robot manipulator equipped with a camera, mounted in an eye-in-hand configuration. The goal is to move the robot in order to reach the desired pose of the camera with respect to an object. We assume that the internal parameters of the camera are known, while the pose of the camera with respect to the robot flange is *unknown*, except for some approximate bounds on the rotation and translation components; i.e., we get rid of hand–eye calibration (HEC). We exploit the fact that the Jacobian of the camera displacement is included in a suitable polytope. Thus we can adopt a Lyapunov-based technique that guarantees asymptotic convergence to the desired camera-to-object reference pose. Our experimental results, performed on a 6 degrees-of-freedom (DoF) robotic manipulator, show that the proposed approach is effective in both achieving a single target pose and performing target object tracking.

1. Introduction

Position-based visual servoing (PBVS) is a well-known tool for guiding robotic systems using visual feedback. By regulating the error between the desired and actual camera pose in the 3D Cartesian workspace [1], PBVS enables a wide range of tasks including object tracking, grasping, and assembly [2]. An important step in implementing PBVS is hand–eye calibration (HEC) [3,4], a meticulous and often error-prone process that establishes the spatial relationship between the camera and the robot's end-effector [5]. Poor calibration can affect PBVS's widespread adoption in practical applications, leading to suboptimal performance and non-robust or inefficient control [6,7]. Uncalibrated visual servoing methods aim to address these shortcomings by not requiring precise knowledge of the camera's intrinsic and extrinsic parameters or the hand–eye relationship. While these methods have gained attention for image-based visual servoing (IBVS) and homography-based visual servoing (HBVS) [8], they are not suitable for PBVS, which relies on accurate 3D reconstruction for task function definition [9]. Several uncalibrated IBVS and HBVS solutions are available in the literature. In [10], a few IBVS strategies are proposed in the uncalibrated case (unknown robot kinematics and camera parameters); they all rely on exploratory movements for estimating the image Jacobian. A comparison of uncalibrated model-free IBVS methods for small-amplitude movements is reported in [11]. The approach proposed in [12] for the same problem relies on a disturbance observer, but requires a nominal image Jacobian to start from. The technique proposed in [13] is specific to the class of concentric-tube robots, while [14] deals with flexible joints manipulators. In [15], again in the context of IBVS, the estimation of the image Jacobian is based on the computation of a projective homography. In [16], a Model Predictive Control approach is used to enforce constraints, such as visibility constraints and joint limits. Regarding PBVS, to the best of our knowledge, the only approach to uncalibrated PBVS is the one proposed in [17] which, however, is adaptive in nature, and actually *does perform* HEC (albeit online).

To bridge this gap, in this study, we depart from conventional PBVS methods by proposing a novel approach that eliminates the need for HEC and online parameter estimation, effectively functioning as an uncalibrated PBVS method. We consider a robot manipulator with a camera in an *eye-in-hand* configuration, i.e., the camera is attached to the robot flange [18], where the intrinsic camera parameters are

☆ This study was carried out within the PNRR research activities of the consortium iNEST (Interconnected Nord-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.

* Corresponding author.
E-mail addresses: erica.salvato@dia.units.it (E. Salvato), blanchini@uniud.it (F. Blanchini), fenu@units.it (G. Fenu), giulia.giordano@unitn.it (G. Giordano), fapellegrino@units.it (F.A. Pellegrino).

https://doi.org/10.1016/j.robot.2025.105045
Received 4 December 2024; Received in revised form 10 April 2025; Accepted 30 April 2025
Available online 4 June 2025

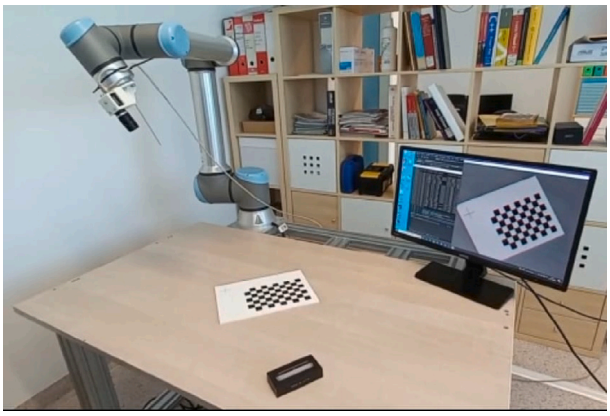

Fig. 1. Our experimental setup. A Universal Robots UR10e manipulator is equipped with a camera in an eye-in-hand configuration. The target object is a checkerboard of known geometry. A servo loop runs in the robot controller (not visible). Target poses are provided by an external PC (whose display, showing the image captured by the camera, is visible), which performs the computations: it estimates the current pose of the target object w.r.t. the camera and solves problem Eq. (36).

known, but the camera's pose relative to the robot flange is unknown, except for approximate bounds; that is, we are not performing HEC. Avoiding the need to perform HEC has several practical advantages in an industrial scenario: indeed, HEC is a time-expensive procedure, since several dozens of images of a calibration pattern must be captured from different camera poses, each obtained with a different robot pose [19]. Moreover, when the camera is dismounted and remounted on the robot flange, the procedure must be repeated. Automating the procedure, so that a set of pre-computed poses of the robot is employed each time the HEC is performed, is not always possible, since the feasibility of the poses depends on the environment, and specifically on obstacles in the surroundings and on possible tools mounted on the robot. Finally, possibly unfavorable lighting conditions, accidental misplacement of the calibration pattern, or other unpredictable factors may compromise the quality of some of the images employed for the HEC. For that reason, the HEC requires, typically, a qualified technician who is able, besides performing the proper procedure, to evaluate the quality of the results. We also stress that, even when HEC can be performed, human errors are possible and our technique successfully compensates for those. Notice that the internal parameters of the camera are the output of a camera calibration procedure (different from the HEC) that is easy to perform, does not require the camera to be moved by the robot nor to be mounted on it, and provides quite accurate final results [20]. It is worth noting that our approach does not rely on machine learning techniques (that may require trial and error attempts) [21–23], nor adaptation [24,25], nor prior data acquisition [26–28], making it highly suitable for plug-and-play applications.

The following are the main contributions of the paper:

- We show that PBVS can be achieved without prior HEC, only exploiting approximate bounds on the components of the tuple defining the unknown camera-to-flange transformation. More specifically, we prove that the general methodology known as Model-Free Plant Tuning (MFPT) [29,30] can effectively solve the PBVS control problem in the presence of uncertainty.[1]
- Given interval uncertainty on the rotation and translation components of the camera pose with respect to the robot flange, we show that (i) the uncertain Jacobian matrix, which links infinitesimal robot-flange displacements to infinitesimal camera displacements, can be embedded in a suitable non-singular polytope of matrices, and (ii) the robust tuning problem can be solved by a dynamic law based on the real-time solution of a quadratic optimization problem on that polytope.
- Our experiments performed on a 6 DoF manipulator show that the technique can successfully handle both single target pose achievements and target object tracking. We compare our approach with a simple gradient descent method that, unlike our methodology, requires full knowledge of the model.

Hereinafter, we denote by I_n the $n \times n$ identity matrix, and by $0_{n\times m}$ the $n \times m$ null matrix. Furthermore, we will specify the unknown pose of the camera with respect to the robot in terms of the rigid transformation from the robot flange to the camera (instead of the flange, the end-effector could be used as a reference).

2. Preliminary results and problem setting

2.1. Model-free plant tuning

For the sake of completeness, we summarize here the main results of [29] that we exploit in the next sections.

Problem 1. Given the static plant

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

where $g : \mathbb{R}^m \to \mathbb{R}^p$, $p \leq m$, is a smooth function, assume that $g(\bar{u}) = 0$ for some *unknown* $\bar{u}$ and that

$$J \doteq \left[\frac{\partial g}{\partial u}\right] \in \mathcal{J}, \tag{2}$$

where J is the Jacobian of g, $\doteq$ is the definition symbol, and $\mathcal{J}$ is a known polytope of matrices. Find a dynamic algorithm such that, as $t \to \infty$,

$$y(t) \to 0, \tag{3}$$

$$u(t) \to \bar{u}, \tag{4}$$

where $\bar{u}$ solves the equation $0 = g(u)$.

The only available information for achieving the goal is thus the inclusion Eq. (2).

Definition 1. A polytope $\mathcal{J}$ is *robustly non-singular* (respectively, *robustly right-invertible*) if all matrices in $\mathcal{J}$ are non-singular (respectively, right-invertible).

We consider the following control scheme.

Theorem 1 ([29]). *If the polytope $\mathcal{J}$ in Eq. (2) is robustly non-singular, or robustly right-invertible, then Problem 1 can be solved by a control scheme of the form:*

$$\dot{u}(t) = v(t), \tag{5}$$

$$v(t) = \Phi(y(t)), \tag{6}$$

$$y(t) = \textit{measured output}, \tag{7}$$

$$v(t) \in \mathcal{V} = \{v : \|v\| \leq \xi(y)\}, \tag{8}$$

where $\xi(y) > 0$ is a continuous, positive and non-decreasing function, while $\|\cdot\|$ is any norm. □

The control can be computed online by solving the convex optimization problem (from now on, we drop the dependency on time, for ease of notation):

$$J^* = \arg\min_{J\in\mathcal{J}} \|y^\top J\|_D, \tag{9}$$

where $\|\cdot\|_D$ is the dual norm of $\|\cdot\|$. Then

$$v = \Phi(y) \doteq -\frac{(J^*)^\top y}{\|(J^*)^\top y\|}\xi(y), \quad y \neq 0. \tag{10}$$

[1] Recently, MFPT has been employed, in a completely different setting, for the control of robots of various kinematic structures [31].

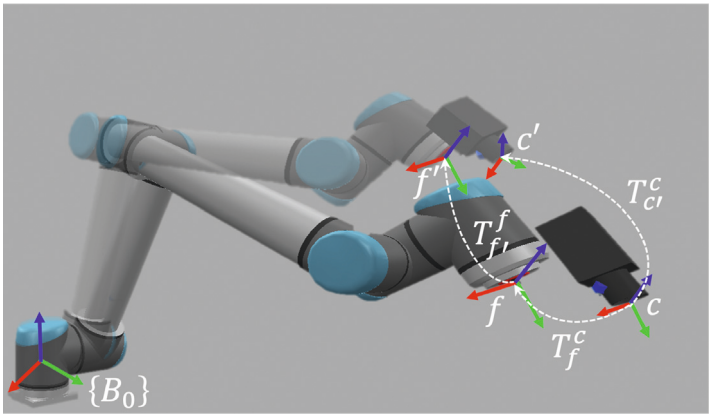

Fig. 2. Representation of the displacement of a robot manipulator in two different flange poses, and the corresponding camera poses.

Theorem 1 states that we can solve Problem 1 if we know a polytopic set $\mathcal{J}$, in the matrix space, in which the Jacobian is confined. Our *main result* is to show how we can determine such a set $\mathcal{J}$, in the considered visual servo control problem, even when we do not have any kind of information about function g.

2.2. Position-based visual servoing

We consider a PBVS setup [32] consisting of a manipulator equipped with a camera attached to the flange. Our goal is to control the manipulator so that the camera is driven to a target pose with respect to an object.

We assume that the pose of the camera with respect to the object can be estimated from a single image captured by the camera. This is a basic assumption of PBVS, in which, for instance, the pose can be estimated via well-known *Perspective-n-Point* (PnP) algorithms [33], which estimate the pose of a camera given a set of n 3D points in the world and their corresponding 2D projections in the image. Moreover, we assume that, while the internal parameters of the camera are known, *the pose of the camera with respect to the flange is unknown*. More precisely, only approximate and/or qualitative information about the pose of the camera with respect to the flange is available; for instance, we may know that "the roll angle is positive and less than 90°".

In the following, the input is the rate of change $v = \dot{u}$ of the parameters representing the pose of the flange with respect to a world reference frame, e.g., the robot base $\{\mathcal{B}_0\}$ of Fig. 2. We neglect dynamics and assume that the robot controller can perfectly track the reference signal, although the model-free plant tuning approach can deal with parasitic dynamics as well [34].

We denote by T^i_j the transformation matrix with respect to the frame i, yielding frame j:

$$T^i_j = \begin{bmatrix} R^i_j & t^i_j \\ 0_{3\times1} & 1 \end{bmatrix}, \tag{11}$$

where R^i_j is the rotation matrix and t^i_j is the translation vector of the frame j with respect to the frame i.

We now consider the robotic system depicted in Fig. 2, where a flange of the manipulator is in the pose f, corresponding to the camera pose c (black), and we want to properly choose the next flange pose f' to drive the camera to the desired pose c' (gray). The transformation matrix T^f_c of the camera with respect to the manipulator flange is fixed but uncertain. By composing the several transformations involved, we get

$$T^c_{c'} = T^c_f T^f_{f'} T^{f'}_{c'} = \left(T^f_c\right)^{-1} T^f_{f'} T^{f'}_{c'}, \tag{12}$$

where $T^f_{f'}$ is the sought transformation matrix of f' with respect to f. If T^f_c were known exactly, Eq. (12) could be solved for $T^f_{f'}$, i.e., the displacement of the manipulator relative to its current pose that results in the required relative displacement of the camera $T^c_{c'}$. However, as mentioned before, this is not our case. We now express the relative pose of a reference frame r' with respect to the frame r, in terms of translations $(x^r_{r'}, y^r_{r'}, z^r_{r'})$ and yaw-pitch-roll angles $(\alpha^r_{r'}, \beta^r_{r'}, \gamma^r_{r'})$. The pose vector of r' with respect to r is thus defined as

$$X^r_{r'} = \begin{bmatrix} x^r_{r'} \\ y^r_{r'} \\ z^r_{r'} \\ \alpha^r_{r'} \\ \beta^r_{r'} \\ \gamma^r_{r'} \end{bmatrix} \doteq \begin{bmatrix} t^r_{r'} \\ \Theta^r_{r'} \end{bmatrix},$$

where $t^r_{r'}$ and $\Theta^r_{r'}$ denote the translation and rotation parts, respectively. The pose change of the camera $X^c_{c'}$, determined by that of the manipulator flange $X^f_{f'}$, is given by

$$X^c_{c'} = \tilde{g}\left(X^f_{f'}\right), \tag{13}$$

where $\tilde{g}(\cdot)$ is independent of the actual configuration of the manipulator and only depends on T^f_c. Since the manipulator kinematics is known exactly, $X^f_{f'}$ can be easily imposed. Thus, in principle, given the desired camera displacement $\bar{X}^c_{c'}$, we could solve Eq. (13) for the corresponding flange displacement $\bar{X}^f_{f'}$, i.e., the flange displacement that satisfies:

$$\bar{X}^c_{c'} = \tilde{g}\left(\bar{X}^f_{f'}\right), \tag{14}$$

and apply that flange displacement in a single shot. However, $\tilde{g}$ is not known, since it depends on the unknown T^f_c. Thus, we exploit the dynamic algorithm provided by MFPT to let $X^c_{c'} \to \bar{X}^c_{c'}$ as $X^f_{f'} \to \bar{X}^f_{f'}$. By subtracting (14) to (13), and letting $g \doteq \tilde{g} - \bar{X}^c_{c'}$, we get

$$E^c_{c'} \doteq X^c_{c'} - \bar{X}^c_{c'} = g\left(X^f_{f'}\right), \tag{15}$$

where we denote by $E^c_{c'}$ the error to be driven to zero, i.e., the difference between the actual and desired camera pose. Letting $u = X^f_{f'}$, the signal to be computed by the algorithm is its rate of change $v = \dot{u}$.

Our approach is framed within the PBVS, and our error is therefore defined in the Cartesian space. Conversely, in IBVS and HBVS control approaches the error is defined differently. IBVS computes the control error in the image plane, typically relying on feature extraction from images, while HBVS approaches use a combination of both image plane and Cartesian space information.

3. Model-free plant tuning for PBVS: determining $\mathcal{J}$

To solve the PBVS within the MFPT framework, we must show that the Jacobian is confined in a polytope $\mathcal{J}$.

By differentiating Eq. (15) with respect to $X^f_{f'}$, we get

$$\begin{bmatrix} dt^c_{c'} \\ d\Theta^c_{c'} \end{bmatrix} = J \begin{bmatrix} dt^f_{f'} \\ d\Theta^f_{f'} \end{bmatrix}, \tag{16}$$

where the 6×6 Jacobian matrix can be written as [35]

$$J = \begin{bmatrix} \frac{\partial t^c_{c'}}{\partial t^f_{f'}} & \frac{\partial t^c_{c'}}{\partial \Theta^f_{f'}} \\ \frac{\partial \Theta^c_{c'}}{\partial t^f_{f'}} & \frac{\partial \Theta^c_{c'}}{\partial \Theta^f_{f'}} \end{bmatrix} = \begin{bmatrix} \left(R^f_c\right)^\top & -\left(R^f_c\right)^\top \left[t^f_c\right]_\times \\ 0_{3\times 3} & \left(R^f_c\right)^\top \end{bmatrix}, \tag{17}$$

where

$$R^f_c = \begin{bmatrix} c_\alpha c_\beta & -c_\alpha s_\beta & s_\alpha \\ s_\gamma s_\alpha c_\beta + c_\gamma s_\beta & -s_\gamma s_\alpha s_\beta + c_\gamma c_\beta & -s_\gamma c_\alpha \\ -c_\gamma s_\alpha c_\beta + s_\gamma s_\beta & c_\gamma s_\alpha s_\beta + s_\gamma c_\beta & c_\gamma c_\alpha \end{bmatrix}, \tag{18}$$

$c_\phi \doteq \cos\phi$, $s_\phi \doteq \sin\phi$, $\alpha = \alpha^f_c$, $\beta = \beta^f_c$ and $\gamma = \gamma^f_c$, and by $[p]_\times$ we denote the skew-symmetric matrix that encodes in matrix form the cross product operation with the vector p:

$$[p]_\times = \begin{bmatrix} 0 & p_1 & -p_2 \\ -p_1 & 0 & p_3 \\ p_2 & -p_3 & 0 \end{bmatrix}. \tag{19}$$

We first consider the uncertainty in the angles α, β, γ; the uncertainties in the components of the translation vector t^f_c affecting $\left[t^f_c\right]_\times$ will be considered afterward. The matrix defined in Eq. (18) has no polytopic structure with respect to its parameters α, β, γ, thus J has no polytopic structure as well. As a consequence, we cannot directly apply Theorem 1. However, as shown below, we can conveniently embed (18) and, ultimately, (17) into a suitable polytopic family, thus allowing the application of the model-free plant tuning technique. We begin by introducing the bounds[2]

$$0 < \alpha^- \le \alpha \le \alpha^+ < \pi/2, \tag{20}$$

$$0 < \beta^- \le \beta \le \beta^+ < \pi/2, \tag{21}$$

$$0 < \gamma^- \le \gamma \le \gamma^+ < \pi/2. \tag{22}$$

and the auxiliary variables

$$q_1 \doteq c_\alpha,\ q_2 \doteq c_\beta,\ q_3 \doteq c_\gamma, q_4 \doteq s_\alpha,\ q_5 \doteq s_\beta,\ q_6 \doteq s_\gamma.$$

Then, we observe that

$$\begin{aligned} &c_{\alpha^+} \le q_1 \le c_{\alpha^-}, \quad s_{\alpha^-} \le q_4 \le s_{\alpha^+}, \\ &c_{\beta^+} \le q_2 \le c_{\beta^-}, \quad s_{\beta^-} \le q_5 \le s_{\beta^+}, \\ &c_{\gamma^+} \le q_3 \le c_{\gamma^-}, \quad s_{\gamma^-} \le q_6 \le s_{\gamma^+}. \end{aligned}$$

Hence, the new variables are included in the box

$$\mathcal{Q} = \left\{ q : \ q^- \le q \le q^+ \right\}.$$

With slight abuse of notation, we denote by $R^f_c(q)$ the over-parameterized matrix

$$R^f_c(q) = \begin{bmatrix} q_1 q_2 & -q_1 q_5 & q_4 \\ q_6 q_4 q_2 + q_3 q_5 & -q_6 q_4 q_5 + q_3 q_2 & -q_6 q_1 \\ -q_3 q_4 q_2 + q_6 q_5 & q_3 q_4 q_5 + q_6 q_2 & q_3 q_1 \end{bmatrix} \tag{23}$$

obtained by substituting the auxiliary variables into Eq. (18): we deliberately disregard the dependence among the components of q (i.e., the coupling between sine and cosine) with the fundamental advantage that the family described by (23) is multi-affine [36] with respect to the variables q_i.

Let $\left\{R_1,\ R_2,\ \ldots,\ R_{2^6}\right\}$ denote the set of matrices $R^f_c(q)$ corresponding to q taken on the 2^6 vertices of the box $\mathcal{Q}$; for instance, $q = \left[q_1^+, q_2^+, q_3^-, q_4^+, q_5^+, q_6^-\right]^\top$ is one of these vertices. Consider the convex polytope $\mathcal{R}$ obtained as the convex hull of matrices R_k:

$$\mathcal{R} = \text{conv}\left\{R_k,\ k = 1, 2, \ldots, 2^6\right\}. \tag{24}$$

Such a polytope will be shown to play a fundamental role when applying model-free plant tuning to visual servoing; for now, we observe that all the rotation matrices R^f_c appearing in Eq. (17), and satisfying Eq. (20)–Eq. (22), are included in $\mathcal{R}$.

We now consider the translation uncertainty and assume that the components of t^f_c are subject to the bounds[3]

$$x^{f-}_c \le p_1 \doteq x^f_c \le x^{f+}_c, \tag{25}$$

$$y^{f-}_c \le p_2 \doteq y^f_c \le y^{f+}_c, \tag{26}$$

$$z^{f-}_c \le p_3 \doteq z^f_c \le z^{f+}_c. \tag{27}$$

Then, in view of Eq. (19), both matrices $\left(R^f_c(q)\right)^\top$ and $\left[t^f_c(p)\right]_\times$ have a multi-affine structure in the variables $q_1, \ldots, q_6$ and p_1, p_2, p_3 respectively.

We denote by $\mathcal{T}$ the polytope in the three-dimensional space of the skew-symmetric matrices Eq. (19) corresponding to p as in Eqs. (25)–(27):

$$\left[t^f_c\right]_\times \in \mathcal{T} = \left\{ p_1 \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + p_2 \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} + p_3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix},\ p_k \text{ as in Eq.(25)-Eq.(27)} \right\}.$$

We now report a simple, yet fundamental, technical result [36].

Proposition 1. *The product $\Pi(q,p) = M(q)N(p)$ of two generic multi-affine matrices $M(q)$ and $N(p)$, with independent parameters p and q, is multi-affine.*

Proof. Any entry of $\Pi_{ij} = \sum_k M_{ik}N_{kj}$ involves the sum of products $M_{ik}(q)N_{kj}(p)$. The product of two multi-affine functions of independent parameters is multi-affine and the sum of multi-affine functions is multi-affine as well. □

As a consequence, matrix J in Eq. (17) is multi-affine:

$$J(q,p) = \begin{bmatrix} (R^f_c(q))^\top & 0_{3\times 3} \\ 0_{3\times 3} & (R^f_c(q))^\top \end{bmatrix} \begin{bmatrix} I_3 & -t^f_c(p) \\ 0_{3\times 3} & I_3 \end{bmatrix}. \tag{28}$$

We now recall the fundamental Mapping Theorem [36].

Theorem 2 (*Mapping Theorem*). *Let $f : \mathbb{R}^d \to \mathbb{R}^e$ be a multi-affine function defined on a hyper-rectangle $\mathcal{Q}$. Then, the image $f(\mathcal{Q})$ is a subset of the convex hull of all the points $f(q_k)$, where q_k are the vertices of the hyper-rectangle $\mathcal{Q}$.*

As a consequence, the Jacobian belongs to a polytope $\mathcal{J}$. Let $R_k \in \text{vert}\mathcal{R}$, $k = 1, 2 \ldots, 32$, and $\left[t_i\right]_\times \in \text{vert}\mathcal{T}$, $i = 1, 2 \ldots 8$. Then the including polytope is

$$\mathcal{J} = \text{conv}\left\{ M_{k,i} \doteq \begin{bmatrix} R_k^\top & -R_k^\top \left[t_i\right]_\times \\ 0_{3\times 3} & R_k^\top \end{bmatrix} \right\} \tag{29}$$

having $2^6 \cdot 2^3 = 2^9 = 512$ vertices.

Note that the inclusion is in general strict: there are points in $\mathcal{J}$ that do not correspond to possible values of J, although the polytope, being the convex hull, is the smallest one in which we can confine the Jacobian set.

[2] While 0 and $\pi/2$ are the theoretical bound values for which the theory is fully applicable, the internal bounds $\alpha^-, \alpha^+, \ldots$ can be chosen arbitrarily based on the specific application. To further extend the $\pi/2$ limit, one can, in practice, change the reference frame during the operation.

[3] Also in this case, the bounds $x^{f-}_c, x^{f+}_c, \ldots$ can be chosen arbitrarily, depending on the specific application in which the approach is used.

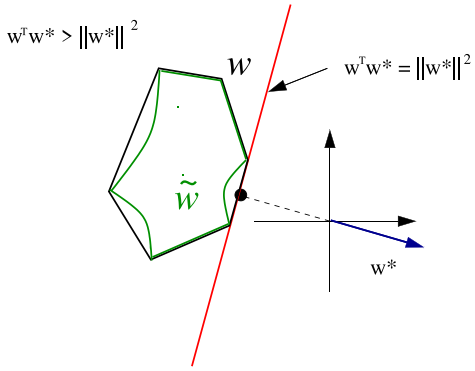

Fig. 3. The separating hyperplane (red): for all $w \in \mathcal{W}$, the inequality $w^\top w^* \geq \|w^*\|^2$ holds. Set $\mathcal{W}$ is the convex hull of the "true" set $\tilde{\mathcal{W}}$ of vectors $M^\top y$. A separating hyperplane for the true set is also a separating hyperplane for the convex hull, hence no conservativeness is introduced. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Now, by defining $y \doteq E_c^c$, and $u \doteq X_f^f$, our PBVS problem Eq. (15) can be cast into the MFPT formulation $\dot{y} = g(u)$, so driving the camera to the desired pose has the same form as Problem 1. We can now state the following result.

Theorem 3. *Assume that the convex polytope $\mathcal{R}$ is robustly non-singular, or equivalently that the polytope $\mathcal{J}$ defined in Eq. (29) is robustly non-singular. Then, a control $\dot{u} = v$ that guarantees $y \to 0$ as $t \to \infty$ can be chosen as*

$$v = -\eta w^*, \tag{30}$$

where $\eta > 0$ is a gain, and $w^* = (M^*)^\top y$ is the (unique) point with minimum Euclidean distance from the origin:

$$M^* = \arg\min_{M \in \mathcal{J}} \left\| y^\top M \right\|_2. \tag{31}$$

Proof. The equivalence between the robust non-singularity of $\mathcal{R}$ and of $\mathcal{J}$ immediately follows from the block-triangular form of J in Eq. (17). To prove the main claim, consider the differential inclusion

$$\dot{y}(t) = M(t)v(t), \qquad M(t) \in \mathcal{J}, \tag{32}$$

and the Lyapunov function $V(y) = \frac{1}{2} y^\top y$. For the generic y, compute $\dot{V} = y^\top M v \doteq w^\top v$, where the vector $w = M^\top y$ is included in the polytope of vectors $\mathcal{W} \doteq \{w : w = M^\top y, \ M \in \mathcal{J}\}$. This polytope is the convex hull of all possible vectors $w = J(p,q)^\top y$, with J being a multi-affine function of the parameters. The distance of the polytope from the origin

$$\mu(y) = \min \|w\| > 0, \quad w \in \mathcal{W},$$

is strictly positive; in fact, $M^\top y = w = 0$ would contradict robust non-singularity. Now, let w^* be the (unique) vector having minimum distance $\|w^*\| = \mu(y)$; then

$$\left\|w^*\right\| = \mu(y) \geq \mu_0 \|y\|, \tag{33}$$

where

$$\mu_0 \doteq \min_{\|y\|=1} \min_{M \in \mathcal{J}} \left\| y^\top M \right\| > 0.$$

The existence of the minimum follows from the fact that the minimal selection $\phi_{min}(y) = \min_{M \in \mathcal{J}} \|y^\top M\|$ is a continuous function of y [37], and hence it achieves its minimum μ_0 on the unit sphere $\|y\| = 1$, which is a compact set. The statement is thus proven just by scaling, in view of the linearity of $w = M^\top y \in \mathcal{W}$ with respect to y.

From the duality theorem in [38, Section 5.13], there exists a separating hyperplane, passing through w^* and having equation $w^\top w^* = \|w^*\|^2$, for which $w^\top w^* \geq \|w^*\|^2$, for all $w \in \mathcal{W}$; see Fig. 3. Then, from Eq. (33),

$$-w^\top w^* \leq -\|w^*\|^2 \leq -\mu_0^2 \|y\|^2, \quad \forall \ w \in \mathcal{W}.$$

Consider $v = -\eta w^*$, as in Eq. (30), and recall that $M^\top y = w$. Then, for all $M \in \mathcal{J}$,

$$\dot{V}(y) = y^\top M v = y^\top M(-\eta w^*) = -\eta w^\top w^* \leq -\eta \mu_0^2 \|y\|^2. \tag{34}$$

In view of this Lyapunov condition, our control v Eq. (30) ensures the convergence to zero of the state of the differential inclusion Eq. (32), not of the original system. However, the Mapping Theorem [36] guarantees that the Jacobian of the original system is actually included in the convex hull $\mathcal{J}$ defined in Eq. (29). Hence the trajectories of the original system are a subset of those of the differential inclusion Eq. (32), and therefore Eq. (34) holds for the original system as well. □

Remark 1. While any positive η ensures convergence, which is as fast as $\exp(-\eta\mu_0^2 t)$ according to Eq. (34), the precise choice of η depends on the trade-off between the desired convergence speed and the acceptable exploitation level of the actuator. Alternatively, a high gain adaptive control [39] can be adopted, in which the gain is progressively increased until convergence is achieved (within a prescribed arbitrarily small tolerance, see [40] for details).

Remark 2. At first sight, the bounds (20)–(22) and (25)–(27) may seem to restrict the applicability of the approach. However, this is not the case, since the flange reference frame is essentially arbitrary. In other words, the reference frame f can be *virtual*, i.e., it can be *chosen* in order to satisfy the bounds on the angles and translation that define T_c^f. For example, small rotations of the virtual reference frame with respect to the actual flange frame may guarantee that the roll, pitch, and yaw angles are positive. This is exactly what has been done in the experiments reported in Section 5.

Remark 3. The members of the family defined in (18) belong, by construction, to the special orthogonal group SO(3). On the other hand, the convex polytope described in (24) is a superset of (18), and, clearly not all of its members belong to SO(3). However, this is by no means a problem as long as the Jacobian belongs to a non-singular polytope.

A key point in the proof of Theorem 3 is the existence of a separating hyperplane, which is equivalent to the robust non-singularity of the polytope $\mathcal{J}$. To comment on this, we remind a property of separating hyperplanes.

Proposition 2. *Let $w(p)$ be a multi-affine vector function, with p belonging to a hyper-rectangle; denote by $\tilde{\mathcal{W}}$ its image set and by $\mathcal{W}$ the convex hull of $\tilde{\mathcal{W}}$. Then a separating hyperplane, passing through w^* and having equation $w^\top w^* = \|w^*\|^2$, for which $w^\top w^* \geq \|w^*\|^2$, exists for all $w \in \tilde{\mathcal{W}}$ if and only if such a hyperplane exists for all $w \in \mathcal{W}$.*

In view of the proposition above, the non-singularity check that we perform to ensure the existence of the separating hyperplane must be performed on the whole convex hull $\mathcal{J}$, not on the original set of all possible Jacobians $J(p,q)$.

It is worth pointing out that our framework allows us to also deal with the case in which the reference r is moving, instead of being fixed. Consider the model

$$y(t) = g(u(t)) - r(t), \tag{35}$$

where we can measure the error y, but we do not know r and its speed $\dot{r}$. The only available information is that the unknown speed satisfies the bound $\|\dot{r}(t)\| \leq s_{max}$, where s_{max} is known, at all times t.

Although we cannot achieve convergence to zero, we can achieve practical convergence to zero [41], namely, convergence within an arbitrarily small ball.

Proposition 3. *Consider system (35), where g satisfies the same assumptions previously considered, r is unknown and its speed is also unknown but bounded as $\|\dot{r}(t)\| \leq s_{max}$, $\forall t$. Then, practical stability can be achieved, namely, the measured output $y(t)$ converges to the ball with radius $s_{max}/(\eta\mu_0^2)$:*

$$|y| \leq \frac{s_{max}}{\eta\mu_0^2}.$$

The radius can be made arbitrarily small by selecting a suitably large gain η.

Proof. The same computation performed to derive (34) can be repeated and the resulting derivative contains an additional term:

$$\begin{aligned}\dot{V} &= -\eta\mu_0^2\|y\|^2 - y^\top\dot{r} \leq -\eta\mu_0^2\|y\|^2 + |y^\top\dot{r}| \leq \\ &\quad -\eta\mu_0^2\|y\|^2 + \|y\|\,\|\dot{r}\| \leq \left[-\eta\mu_0^2\|y\| + s_{max}\right]\|y\|\end{aligned}$$

Then, $\dot{V}(y(t)) < 0$ as long as $\|y(t)\| > \frac{s_{max}}{\eta\mu_0^2}$. By adopting standard Lyapunov arguments, (see [41], Section 3.3) this means that $V(y(t))$ is decreasing as long as y is outside the ball, and hence $y(t)$ must converge to such a ball. □

In practice, unmodeled non-idealities, communication and actuation delays, and high-frequency noise may prevent the robot from accurately tracking the reference signal, when the target object moves rapidly.

4. Implementation and non-singularity check

4.1. Real-time implementation of the control

The algorithm to find w^*, and hence compute the control as in (30), is a quadratic programming problem. By letting the index k span the whole set of vertices of $\mathcal{J}$, and denoting by M_k the k-th vertex, the generic matrix belonging to $\mathcal{J}$ can be written as the convex combination $M = \sum_{k=1}^{2^9} M_k\xi_k$, with $\xi_k \geq 0$ and $\sum_{k=1}^{2^9}\xi_k = 1$. Hence,

$$\left\|y^\top M\right\|_2 = \left\|\sum_{k=1}^{2^9} y^\top M_k\xi_k\right\|_2 = \left\|\Omega(y)^\top\xi\right\|_2,$$

where $\xi = \left[\xi_1, \xi_2, \ldots \xi_{2^9}\right]^\top$ and $\Omega(y)^\top \doteq \left[M_1^\top y,\ M_2^\top y,\ \ldots,\ M_{2^9}^\top y\right]$. Then, the problem to be solved (online) is

$$\begin{aligned}\xi^* = \arg\min\ & \xi^\top\Omega(y)\Omega(y)^\top\xi, \\ \text{s.t.}\ & \xi_k \geq 0,\ \sum_{k=1}^{2^9}\xi_k = 1.\end{aligned} \tag{36}$$

Then Eq. (30), namely $v = -\eta w^*$, holds with

$$w^{*\top} = \sum_{k=1}^{2^9} y^\top M_k\xi_k^*. \tag{37}$$

Remark 4. In view of Eq. (34), parameter $\eta > 0$, associated with the convergence speed, is arbitrary. However, large values of η require an implementation with a small sampling time, which in turn is lower bounded by the time needed to compute the control. A too-large value of η could thus introduce chattering and even instability.

4.2. Offline computation: non-singularity check

Checking the robust non-singularity of $\mathcal{J}$ requires checking the robust non-singularity of $\mathcal{R}$. Checking robust singularity is a hard problem in general [36], but is reasonably simple in our case, according to the following proposition.

Proposition 4. *The set $\mathcal{R}$ defined in Eq. (24) is robustly non-singular if and only if*

$$v_0 = \min_{\|z\|=1,\ R\in\mathcal{R}} \left\|z^\top R\right\| > 0.$$

The minimum v_0 is well defined in view of the continuity of the minimal selection function [37]. To determine such a minimum, we parameterize the unit vectors z as $z(\psi_1,\psi_2) \doteq [c_{\psi_1}c_{\psi_2},\ s_{\psi_1}c_{\psi_2},\ s_{\psi_2}]^\top$ and, for all $0 \leq \psi_1 \leq 2\pi$ and $-\pi/2 \leq \psi_2 \leq \pi/2$, we solve (offline and by discretization) the parametric quadratic problem

$$\begin{aligned}\rho(\psi_1,\psi_2) = \min\ & \zeta^\top\Psi(z)\Psi(z)^\top\zeta, \\ \text{s.t.}\ & \zeta_k \geq 0,\ \sum_{k=1}^{2^6}\zeta_k = 1,\end{aligned} \tag{38}$$

where $\Psi(z)^\top \doteq \left[R_1^\top z,\ R_2^\top z,\ \ldots,\ R_{2^6}^\top z\right]$. Then, we take

$$v_0 = \min_{0\leq\psi_1\leq 2\pi,\ \pi/2\leq\psi_2\leq\pi/2} \rho(\psi_1,\psi_2), \tag{39}$$

and check the condition $v_0 > 0$.

Remark 5. The quantity μ_0 in Eq. (34) can be determined in a similar way as v_0, with the additional difficulty that the unit sphere to be considered is in the 6-dimensional space, which can be parameterized in a similar way. Adopting 5 angles ψ_k, the unit vector has components

$$\left[\ c_1c_2c_3c_4c_5,\ \ s_1c_2c_3c_4c_5,\ \ s_2c_3c_4c_5,\ \ s_3c_4c_5,\ \ s_4c_5,\ \ s_5\right]$$

where $c_k = \cos(\psi_k)$ and $s_k = \sin(\psi_k)$. The parameterized optimization has to be performed in dimension 5.

The following procedures summarize the steps necessary to synthesize and apply the control law. Note that the on-line procedure refers to a discrete-time implementation, where τ is the sampling time.

Procedure 1. (Off-line)
Given the bound (20)–(22), perform the following operations:

1. *Compute the bounds on the components of q.*
2. *Grid the square $0 \leq \psi_1 \leq 2\pi$ and $-\pi/2 \leq \psi_2 \leq \pi/2$.*
3. *According to Proposition 4, solve the linear quadratic problem (38) for each point in this grid.*
4. *To perform the non-singularity check, compute the overall minimum v_0 in (39) and check if $v_0 > 0$. If the condition is false, the test fails, and hence STOP. Otherwise, continue.*
5. *Choose parameter $\eta > 0$ according to Theorem 3 and a sampling time $\tau > 0$. Fix an initial input $u(0) = X^f_{f'}(0)$.*

Procedure 2. (On-line)
At each time step:

1. *Measure y.*
2. *Solve the linear-quadratic problem (31) in Theorem 3 to determine M^*.*
3. *Compute $v = -\eta(M^*)^\top y$.*
4. *Integrate v to derive the next u as $u := u + \tau v$*
5. *Apply u and wait until the next time step.*

5. Experimental results

As a first experiment to illustrate the proposed technique, we performed a regulation to a constant target pose. A video of one of the MFPT application experiments can be found at: https://youtu.be/-ZOhOmUE3do. Since the essential feature of the proposed method is its (almost) model-free nature, we decided to compare the obtained trajectories with the ones of a model-aware approach. In particular, we considered the full-information case in which the pose of the camera with respect to the flange (and hence the Jacobian) is known as the result of an HEC procedure. In addition, we decided to systematically analyze the performance of the MFPT approach as far as possible both when varying the camera pose with respect to the flange and

when varying the translation and rotation uncertainties. As a second experiment, we performed a tracking experiment where the object is manually repeatedly displaced in several different positions during the trial, to assess the sensitivity to disturbances. A video of the tracking experiment can be accessed at the URL: https://youtu.be/TfGylkhYaUc. The experimental setup, shown in Fig. 1, consists of a Universal Robots UR10e manipulator, equipped with a Hikrobot MV–CE013–80GM camera, having a resolution of 1280 × 1024 and frame rate of about 20 fps. In our case study, the target object is a checkerboard of known geometry. It is worth stressing that considering a checkerboard is not a limitation, since the approach could be adopted as well with any other target object. A servo loop runs in the robot controller, whose target poses are provided by PC connected through an Ethernet cable and employing the Real-Time Data Exchange protocol provided by Universal Robots. All the computations are performed by the PC and the target pose is expressed as the relative Cartesian displacement of the flange with respect to its current pose. The servo loop keeps the target constant for 8ms, while the PC sends the new target pose as soon as available. On average, processing a frame and computing the new target pose required about 150ms (on a PC equipped with a 12th Gen Intel(R) Core(TM) i7–12700, clock 2.10 GHz). For each acquired frame, the two main processing steps are the estimation of the current pose of the target object w.r.t. the camera, and the solution to the quadratic programming problem Eq. (36). Both steps have been implemented in C++ in the Microsoft Visual Studio 2021 environment. As for the former, we employed the PnP algorithm of release 4.5 of the OpenCV library [42]. As for the latter, we used the C++ library Eiquadprog [43] which allows us to solve quadratic programming problems via the active-set dual method [44]. The pose of the camera with respect to the robot flange has been estimated via the standard HEC procedure provided by OpenCV. We considered the following two camera mountings:

- **CAM-a**: $x_c^f = 0.03\,\text{m}$, $y_c^f = 0.07\,\text{m}$, $z_c^f = 0.03\,\text{m}$, $\alpha_c^f = 0.09\,\text{rad}$, $\beta_c^f = 0.27\,\text{rad}$, $\gamma_c^f = 0.09\,\text{rad}$, corresponding to the matrix:

$$T_c^f = \begin{bmatrix} 0.95987 & -0.08662 & 0.26673 & 0.03 \\ 0.1134 & 0.98977 & -0.086627 & 0.07 \\ -0.25650 & 0.11339 & 0.95987 & 0.03 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

- **CAM-b**: the rotation angles were doubled $\alpha_c^f = 0.18\,\text{rad}$, $\beta_c^f = 0.54\,\text{rad}$, $\gamma_c^f = 0.18\,\text{rad}$ and the matrix T_c^f updated accordingly.

Clearly, the matrix T_c^f is unknown to our controller and the following ranges have been employed, specific for our experimental system

- **RPY-a**: $\alpha^- = \beta^- = \gamma^- = 0.017\,\text{rad} = 1°$, $\alpha^+ = \beta^+ = \gamma^+ = 0.524\,\text{rad} = 30°$
- **RPY-b**: $\alpha^- = \beta^- = \gamma^- = 0.017\,\text{rad} = 1°$, $\alpha^+ = \beta^+ = \gamma^+ = 0.785\,\text{rad} = 45°$
- **RPY-c**: $\alpha^- = \beta^- = \gamma^- = 0.017\,\text{rad} = 1°$, $\alpha^+ = \beta^+ = \gamma^+ = 1.047\,\text{rad} = 60°$

for the angular uncertainty bounds. Moreover, the following ranges for the translation uncertainties have been employed

- **XYZ-a** $x_c^{f-} = y_c^{f-} = z_c^{f-} = 0.001\,\text{m}$, $x_c^{f+} = y_c^{f+} = z_c^{f+} = 0.1\,\text{m}$.
- **XYZ-b** $x_c^{f-} = y_c^{f-} = z_c^{f-} = 0.001\,\text{m}$, $x_c^{f+} = y_c^{f+} = z_c^{f+} = 0.5\,\text{m}$.

By combining different camera mountings and different uncertainty ranges (both angular and translational), we designed six experimental scenarios, as detailed in Table 1. Among these, Scenarios 3 and 6 exhibit the highest angular uncertainty, with roll, pitch, and yaw angles ranging from 1° to 60°.

The experimental protocol is as follows.

Table 1
The MFPT scenarios in the regulation experiments.

Exp. label	HEC cfg.	RPY cfg.	XYZ cfg.
MFPT-1	CAM-a	RPY-a	XYZ-a
MFPT-2	CAM-a	RPY-b	XYZ-a
MFPT-3	CAM-a	RPY-c	XYZ-a
MFPT-4	CAM-a	RPY-b	XYZ-b
MFPT-5	CAM-b	RPY-b	XYZ-a
MFPT-6	CAM-b	RPY-c	XYZ-a

- The robot is manually positioned in the desired pose with respect to the object (the checkerboard), chosen in such a way that the checkerboard is visible to the camera. The relative pose is recorded in order to allow the computation of the error during the experiment.
- The robot is manually moved to a different pose with respect to the object. Such a pose is chosen in such a way that the checkerboard is visible to the camera, and is the starting pose for the regulation experiment.
- The control (either model-free plant tuning or gradient-based) is turned on and kept running until convergence (the tolerances for the convergence condition are $\left\|[e_x\ e_y\ e_z]^\top\right\| < 0.001\,\text{m}$ and $\left\|[e_\alpha\ e_\beta\ e_\gamma]^\top\right\| < 0.01\,\text{rad}$ for the constant target pose task, while $\left\|[e_x\ e_y\ e_z]^\top\right\| < 0.001\,\text{m}$ and $\left\|[e_\alpha\ e_\beta\ e_\gamma]^\top\right\| < 0.001\,\text{rad}$ for the tracking experiment).

First, we report the results of a regulation experiment, in which we compare the time evolution obtained with our proposed MFPT approach (in the six different scenarios) and with a gradient-based approach, which (differently from our approach) has full knowledge of the system Jacobian matrix Eq. (17). The control law with our proposed approach is given by Eq. (30), with $\eta = 0.5$. In the gradient-based case, the implemented control law is

$$\dot{u} = -\rho J^{-1} y, \tag{40}$$

where the parameter ρ is chosen as $\rho = 0.4$. Fig. 5 reports the components of the error as a function of time. It can be noted that the quality of the transients changes as the level of uncertainty increases, and the settling time is longer for scenarios 3 and 6 (which are particularly challenging, being associated to the highest angular uncertainty). The different camera arrangements have little effect on the transients (compare curves 2 and 5).

A noticeable difference between the proposed method and the gradient-based method can be seen, as expected since the MFPT method considers the worst-case Jacobian, which is not in general the true J that we need to know in order to implement the control Eq. (40). In either case, we can ensure that V is monotonically decreasing. The chattering observed for some variables occurs because η is pushed to the limit given the sampling time, as explained in Remark 4. In Fig. 4, some images acquired during the model-free regulation experiment are reported, along with the corresponding norm of the pose error. The images are superimposed with the image corresponding to the desired pose. We stress that the proposed approach is position-based, thus no comparison between a "desired image" and the actual image is performed; on the contrary, the current image is only used for computing the current pose. Additional results can be found at https://drive.google.com/file/d/1XVV_11IaaxZgixxghLAVQqWG2nSEzku7/view?usp=sharing.

The tracking experiment is essentially a regulation experiment in which the target object is manually repositioned during the experiment. We report in Fig. 6 the component-wise error trajectory for a tracking experiment. The desired pose and the assumptions on the pose of the camera with respect to the robot flange are the same as in the regulation experiment MFPT-1 in Table 1. Here, however, the object is

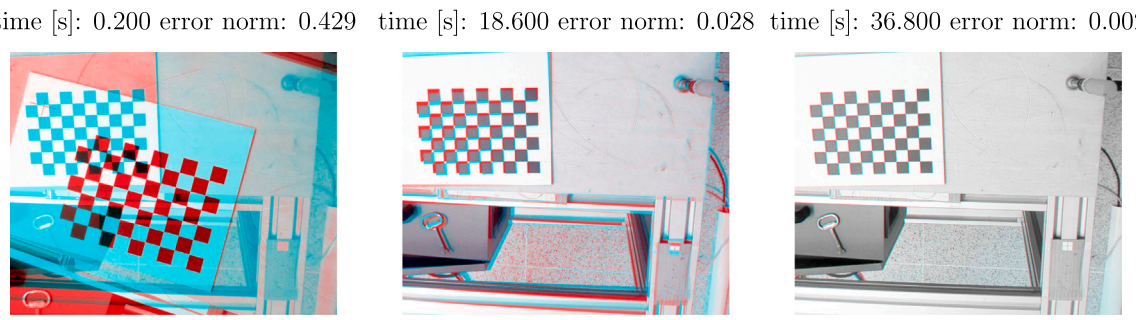

Fig. 4. Some of the images acquired by the camera during the transient (light red) are superimposed with the image corresponding to the desired camera pose (cyan). The error norm reported for each image is the norm of the error vector E_c^c of Eq. (15). The transient corresponds to the experiment MFPT-1, illustrated in Fig. 5. The timestamps associated with each image are the time instants when those frames appear in the full video of the experiment. The full video is available at the URL https://youtu.be/-ZOhOmUE3do. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

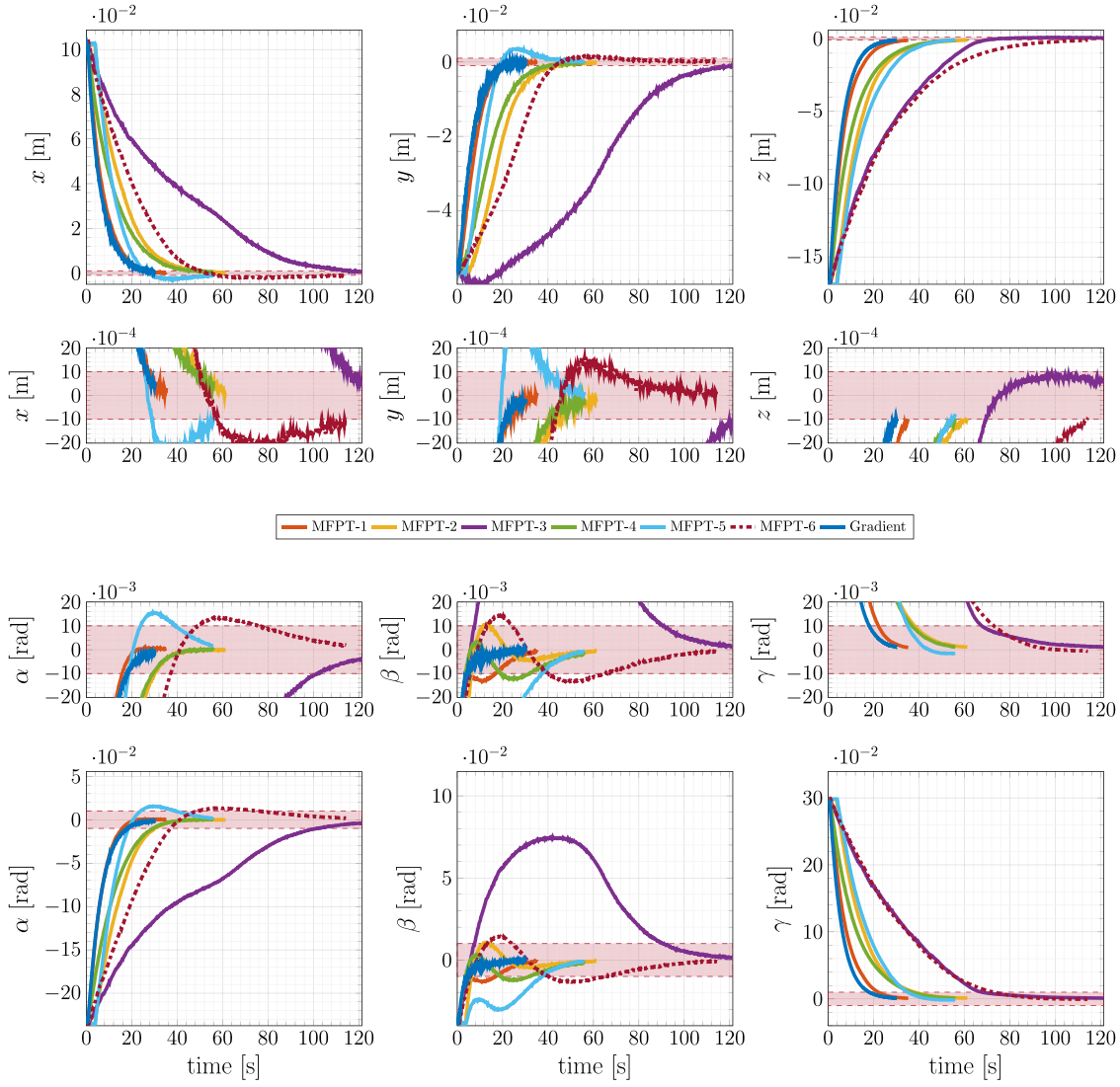

Fig. 5. Component-wise plots of the camera pose error achieved with the full-information gradient-based approach (blue) and with the model-free plant tuning approach (six different scenarios, according to Table 1), for the constant target pose experiments described in Section 5. The red bands (in evidence in the subplots corresponding to each plot) represent the thresholds for the target to be considered reached (± 0.01 rad for rotational components, ± 0.001 m for translational ones). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

repeatedly manually translated and rotated while our MFPT algorithm is running. Thus, the robot adapts its pose to the new object pose. The arrows reported in Fig. 6 represent the instants when the target object is moved.

6. Conclusion

A model-free plant tuning technique has been employed to solve a PBVS problem for a robotic manipulator having a camera mounted in

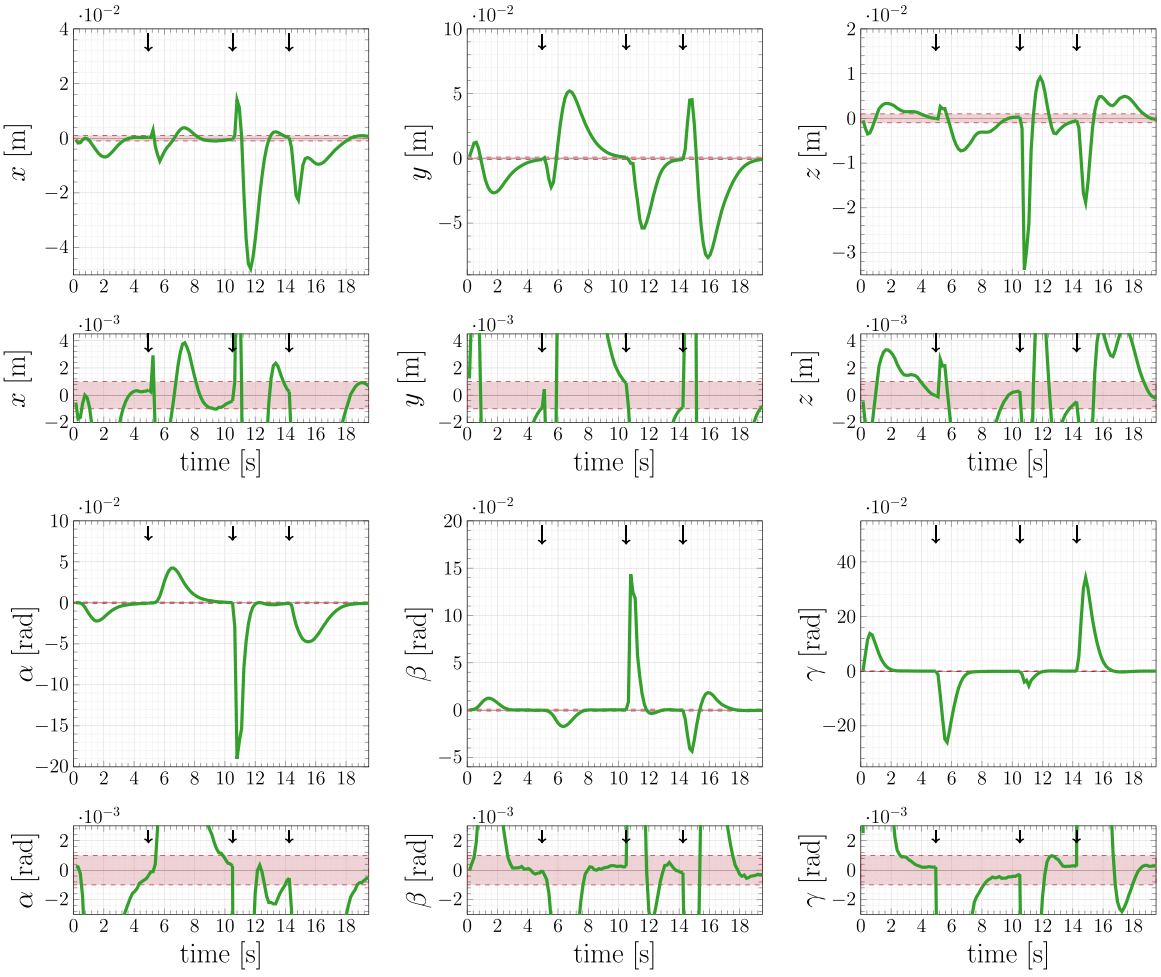

Fig. 6. Component-wise plots of the camera pose error for the tracking experiment. The black arrows represent the time instances in which the position of the object to be tracked is manually changed (after meeting the convergence condition). The red bands (in evidence in the subplots corresponding to each plot) represent the thresholds for the target to be considered reached (± 0.001 rad for rotational components, ± 0.001 m for translational ones). A video of the tracking experiment can be accessed at the URL: https://youtu.be/TfGylkhYaUc. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

an eye-in-hand configuration, without the need to perform HEC. Our solution ensures robust convergence when only approximate information on the camera pose is available. In fact, our proposed robust PBVS controller can effectively handle potential inaccuracies from HEC or even operate without it.

The proposed approach is substantially different from other uncalibrated approaches, in particular, the PBUVS [17] and IBUVS [45–47]. Indeed, those approaches rely on the online estimation of a Jacobian, e.g., the image Jacobian in the case of IBUVS. Such an estimate relies on incremental data acquisition to be refined during the transient. As a consequence, they are sensitive to the initial guess: a wrong initial parameters guess may lead to an unacceptable transient. Our method, instead, is inherently robust, because it does not rely on an online estimate, but is based on a min–max approach. Clearly, the proposed scheme does not guarantee that the target remains in the camera field of view during the whole transient: this is a well-known issue of PBVS, and several techniques to alleviate the problem have been proposed in the literature (see for example [48]). Moreover, we did not explicitly address the issue of kinematic singularities [49–51] that may arise, as our approach operates in Cartesian coordinates. Future work will focus on addressing this challenge while also reducing the complexity of the polytope, including the Jacobian, for a more efficient implementation. Additionally, we plan to develop dedicated software to accelerate computations, enabling higher control gains and faster convergence. Reducing the computation time allows us to increase the control gain and therefore the convergence speed. Finally, the robustness analysis with respect to the camera's intrinsic parameters is a direction for future work.

CRediT authorship contribution statement

Erica Salvato: Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Franco Blanchini:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Gianfranco Fenu:** Writing – review & editing, Visualization, Validation, Investigation, Data curation. **Giulia Giordano:** Writing – review & editing, Formal analysis, Conceptualization. **Felice Andrea Pellegrino:** Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Erica Salvato reports financial support was provided by European Union. Erica Salvato reports a relationship with European Union that includes: funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank Walter Vanzella and Claudia Dorigo of Glance Vision Technologies Srl for providing the experimental setup, for their substantial assistance in performing the experiments, and for the fruitful discussions that contributed to this work.

Data availability

The data that has been used is confidential.

References

[1] V. Lippiello, B. Siciliano, L. Villani, Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration, IEEE Trans. Robot. 23 (1) (2007) 73–86.
[2] J. Wu, Z. Jin, A. Liu, L. Yu, F. Yang, A survey of learning-based control of robotic visual servoing systems, J. Franklin Inst. 359 (1) (2022) 556–577.
[3] R. Tsai, R. Lenz, A new technique for fully autonomous and efficient 3D robotics hand/eye calibration, IEEE Trans. Robot. Autom. 5 (3) (1989) 345–358, http://dx.doi.org/10.1109/70.34770.
[4] J. Jiang, X. Luo, Q. Luo, L. Qiao, M. Li, An overview of hand-eye calibration, Int. J. Adv. Manuf. Technol. 119 (1) (2022) 77–97.
[5] I. Enebuse, B.K.K. Ibrahim, M. Foo, R.S. Matharu, H. Ahmed, Accuracy evaluation of hand-eye calibration techniques for vision-guided robots, PloS One 17 (10) (2022) e0273261.
[6] P. Meer, C.V. Stewart, D.E. Tyler, et al., Robust computer vision: An interdisciplinary challenge, Comput. Vis. Image Underst. 78 (1) (2000) 1–7.
[7] M. Vincze, G.D. Hager, Robust image processing and position based visual servoing, in: Robust Vision for Vision-Based Control of Motion, Wiley-IEEE Press, 2000, pp. 163–201.
[8] B. Espiau, F. Chaumette, P. Rives, A new approach to visual servoing in robotics, IEEE Trans. Robot. Autom. 8 (3) (1992) 313–326.
[9] A. Shademan, M. Jägersand, Three-view uncalibrated visual servoing, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 6234–6239.
[10] M. Hao, Z. Sun, A universal state-space approach to uncalibrated model-free visual servoing, IEEE/ASME Trans. Mechatronics 17 (5) (2011) 833–846.
[11] J. Musić, M. Bonković, M. Cecić, Comparison of uncalibrated model-free visual servoing methods for small-amplitude movements: A simulation study, Int. J. Adv. Robot. Syst. 11 (7) (2014) 108.
[12] Z. Ma, J. Su, Robust uncalibrated visual servoing control based on disturbance observer, Isa Trans. 59 (2015) 193–204.
[13] X. Yang, J. Wang, S. Song, M.Q.-H. Meng, Model-free and uncalibrated eye-in-hand visual servoing approach for concentric-tube robots, IEEE Trans. Instrum. Meas. 71 (2022) 1–11.
[14] Z. Liu, K. Li, T. Hao, H. Wang, Visual servoing of rigid-link flexible-joint manipulators in the presence of unknown camera parameters and boundary output, IEEE Trans. Syst. Man, Cybern.: Syst. (2023).
[15] Z. Gong, B. Tao, H. Yang, Z. Yin, H. Ding, An uncalibrated visual servo method based on projective homography, IEEE Trans. Autom. Sci. Eng. 15 (2) (2017) 806–817.
[16] Z. Qiu, S. Hu, X. Liang, Model predictive control for constrained image-based visual servoing in uncalibrated environments, Asian J. Control 21 (2) (2019) 783–799.
[17] S. He, Y. Xu, D. Li, Y. Xi, Adaptive control algorithm for uncalibrated position-based visual servoing, in: 2022 41st Chinese Control Conference, CCC, 2022, pp. 2192–2197, http://dx.doi.org/10.23919/CCC55666.2022.9902236.
[18] X. Sun, X. Zhu, P. Wang, H. Chen, A review of robot control with visual servoing, in: 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems, CYBER, IEEE, 2018, pp. 116–121.
[19] C. Steger, M. Ulrich, C. Wiedemann, Machine Vision Algorithms and Applications, John Wiley & Sons, 2018.
[20] Y.-J. Zhang, Camera calibration, in: 3-D Computer Vision: Principles, Algorithms and Applications, Springer Nature Singapore, Singapore, 2023, pp. 37–65.
[21] S. Yang, Y.-S. Ong, Y. Jin, Evolutionary computation in dynamic and uncertain environments, vol. 51, Springer, 2007.
[22] E. Medvet, G. Nadizar, GP for continuous control: teacher or learner? The case of simulated modular soft robots, in: Genetic Programming Theory and Practice XX, Springer, 2024, pp. 203–224.
[23] R.S. Sutton, A.G. Barto, et al., Reinforcement learning, J. Cogn. Neurosci. 11 (1) (1999) 126–134.
[24] H. Wang, Adaptive control of robot manipulators with uncertain kinematics and dynamics, IEEE Trans. Autom. Control 62 (2) (2016) 948–954.
[25] K. Lu, Z. Liu, H. Yu, C.P. Chen, Y. Zhang, A small-gain approach to inverse optimal adaptive control of nonlinear systems with unmodeled dynamics, Automatica 159 (2024) 111360.
[26] F.A. Pellegrino, F. Blanchini, G. Fenu, E. Salvato, Closed-loop control from data-driven open-loop optimal control trajectories, in: 2022 European Control Conference, ECC, 2022, pp. 1379–1384, http://dx.doi.org/10.23919/ECC55457.2022.9838344.
[27] F.A. Pellegrino, F. Blanchini, G. Fenu, E. Salvato, Data-driven dynamic relatively optimal control, Eur. J. Control 74 (2023) 100839, http://dx.doi.org/10.1016/j.ejcon.2023.100839, 2023 European Control Conference Special Issue.
[28] F. Blanchini, F. Dabbene, G. Fenu, F.A. Pellegrino, E. Salvato, Model-free feedback control synthesis from expert demonstration, IEEE Control. Syst. Lett. 7 (2023) 1604–1609, http://dx.doi.org/10.1109/LCSYS.2023.3251575.
[29] F. Blanchini, G. Fenu, G. Giordano, F.A. Pellegrino, Model-free plant tuning, IEEE Trans. Autom. Control 62 (6) (2017) 2623–2634.
[30] F. Blanchini, L. Della Schiava, G. Fenu, G. Giordano, F.A. Pellegrino, E. Salvato, Model-free cable robot control, in: IFAC 2023, 2023.
[31] E. Salvato, F. Blanchini, G. Fenu, G. Giordano, F.A. Pellegrino, Model-free kinematic control for robotic systems, Automatica 173 (3) (2025).
[32] F. Chaumette, S. Hutchinson, Visual servoing and visual tracking, in: B. Siciliano, O. Khatib (Eds.), Handbook of Robotics, Springer, 2008.
[33] R. Szeliski, Computer Vision: Algorithms and Applications, Springer Nature, 2022.
[34] F. Blanchini, G. Fenu, G. Giordano, F.A. Pellegrino, Model-free tuning of plants with parasitic dynamics, in: 2017 56th IEEE Conference on Decision and Control, CDC, IEEE, Melbourne, 2017, pp. 499–504.
[35] R. Horaud, F. Dornaika, B. Espiau, Visually guided object grasping, IEEE Trans. Robot. Autom. 14 (4) (1998) 525–532.
[36] B.R. Barmish, New Tools for Robustness of Linear Systems, Macmillan, 1994.
[37] J.P. Aubin, Viability theory, Systems & Control: Foundations & Applications, Birkhäuser Boston Inc., Boston, MA, 1991, p. xxvi+543.
[38] D.G. Luenberger, Optimization by vector space methods, John Wiley & Sons Inc., New York, New York, USA, 1969, p. xvii+326.
[39] A. Ilchmann, Non-identifier-based high-gain adaptive control, Lecture Notes in Control and Information Sciences, 189, Springer-Verlag, London, 1993.
[40] F. Blanchini, S. Miani, Set-Theoretic Methods in Control, second ed., Birkhäuser Basel, 2015.
[41] Z. Qu, Robust Control of Nonlinear Uncertain Systems, Wiley, New York, 1998.
[42] G. Bradski, The OpenCV Library, Dr. Dobb's J. Softw. Tools (2000).
[43] G. Buondonno, Eiquadprog, 2022, https://github.com/stack-of-tasks/eiquadprog.
[44] D. Goldfarb, A. Idnani, A numerically stable dual method for solving strictly convex quadratic programs, Math. Program. 27 (1) (1983) 1–33.
[45] X. Liang, H. Wang, Y.-H. Liu, B. You, Z. Liu, Z. Jing, W. Chen, Fully uncalibrated image-based visual servoing of 2DOFs planar manipulators with a fixed camera, IEEE Trans. Cybern. 52 (10) (2022) 10895–10908, http://dx.doi.org/10.1109/TCYB.2021.3070598.
[46] Z. Gong, B. Tao, H. Yang, Z. Yin, H. Ding, An uncalibrated visual servo method based on projective homography, IEEE Trans. Autom. Sci. Eng. 15 (2) (2018) 806–817, http://dx.doi.org/10.1109/TASE.2017.2702195.
[47] M. Hao, Z. Sun, A universal state-space approach to uncalibrated model-free visual servoing, IEEE/ASME Trans. Mechatronics 17 (5) (2012) 833–846, http://dx.doi.org/10.1109/TMECH.2011.2131149.
[48] B. Thuilot, P. Martinet, L. Cordesses, J. Gallice, Position based visual servoing: keeping the object in the field of vision, in: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), vol. 2, IEEE, 2002, pp. 1624–1629.
[49] J. Kim, G. Marani, W.K. Chung, J. Yuh, A general singularity avoidance framework for robot manipulators: task reconstruction method, in: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, vol. 5, IEEE, 2004, pp. 4809–4814.
[50] E. Salvato, W. Vanzella, G. Fenu, F.A. Pellegrino, Singularity avoidance for cart-mounted hand-guided collaborative robots: A variational approach, Robotics 11 (4) (2022) http://dx.doi.org/10.3390/robotics11040079.
[51] M. Weyrer, M. Brandstötter, M. Husty, Singularity avoidance control of a non-holonomic mobile manipulator for intuitive hand guidance, Robotics 8 (1) (2019) 14.