Autonomous Robots
https://doi.org/10.1007/s10514-018-9782-x

Parking objects by pushing using uncalibrated visual servoing

Gonzalo López-Nicolás[1] · **Erol Özgür**[2] · **Youcef Mezouar**[2]

Received: 9 December 2016 / Accepted: 9 July 2018

Abstract
Pushing is one of the strategies to perform robotic manipulation when the object is too large or too heavy. Motivated by this, we address the problem of how to push an object on a plane to a target pose with two cooperating robots. The main contribution is a new uncalibrated image-based control scheme that computes the required motion of the object to reach the target pose. Then, as an application of this control scheme, we study the conditions that allow performing the task of pushing the object with two robots. The setup consists of a fixed external uncalibrated camera looking at the workspace where the object and the robots stand. The task is defined with a target image of the object in the desired pose. The proposed control scheme computes the motion commands of the pusher robots and, as a result, they translate and rotate the object by imposing non-holonomic velocity constraints. This yields smooth, continuous and efficient trajectories. The stability of the control scheme is also proven. Experiments illustrate the performance of the control scheme.

Keywords Pushing · Manipulation · Visual servoing

1 Introduction

Manipulation of objects using robots is a classical problem that has attracted the attention of the research community during the last decades (Amor et al. 2014). Robotic systems able to perform efficient and robust manipulation are of great interest in many applications. In a pick and place application, the end-effector of the robot grips the object with a clamp or similar device to perform the task. However, this approach can be useless if the object to be grasped is too large or too heavy (Dogar and Srinivasa 2011). Different manipulation strategies have been proposed in the literature to solve such an application using pulling, pushing, throwing, vibrating, etc. These strategies are denoted as non-prehensile or grasp-less manipulation.

Here, we focus on the problem of moving an object by pushing. There has been extensive research on the fundamental mechanics of pushing (Mason 1986). However, this is still a complex problem that raises different challenges with respect to classical grasping such as planning a pushing trajectory to reach the desired configuration or how to push the object. Classical approaches in manipulation by pushing are usually open-loop methods. In general, these works focus on how to define a sequence of pushes to remove the position uncertainty of an object (Akella and Mason 1998; Brost 1986; Lynch 1992). A main problem is to identify a reliable sequence of pushes to achieve the overall pushing while preventing the object from slipping on the contact (Li and Payandeh 2007; Meriçli et al. 2015). However, the slipping of the object on the contact surface of fences or fingers during pushing has been also considered to define a sequence of elementary operations to remove the pose uncertainties of the objects (Peshkin and Sanderson 1988a, b). Due to the open-loop nature of these approaches, failing to accurately follow the planned sequence of pushes or the presence of unaccounted perturbations might yield slip or rolling problems.

The works discussed in the previous paragraph proposed open-loop schemes, and therefore there was no sensing or feedback to compensate for the accumulating error. Another approach considers measurements under closed-loop schemes to improve the robustness against uncertainties and perturbations. The measured data used for the control

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10514-018-9782-x) contains supplementary material, which is available to authorized users.

✉ Gonzalo López-Nicolás
gonlopez@unizar.es

[1] Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, 50009 Saragossa, Spain

[2] Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, 63000 Clermont-Ferrand, France

Published online: 20 July 2018

input can be for instance the angles of the push (Okawa and Yokoyama 1992), tactile feedback (Lynch et al. 1992), or vision (Gandolfo et al. 1991; Golkar et al. 2009; Salganicoff et al. 1993). Just like these latter works, our approach is based on visual feedback. Usually, these methods consist of two different parts. One is the perception system that estimates the position of the object in the Euclidean space and the other is devoted to performing the pushing actions. In contrast, our approach is an image-based control scheme defined directly in terms of uncalibrated image information and performed without computing the 3-D pose of the object.

The related work on manipulation by pushing can be classified depending on the number of contact types used in pushing. For example, there can be a single point contact (Salganicoff et al. 1993), two point contacts (Balorda and Bajd 1994; Golkar et al. 2009), line contact (Akella and Mason 1999) or multiple point contacts (Rus et al. 1995; Sudsang et al. 2002). Our proposal fits in the category of two point contacts push. We propose to use two robots which push the object coordinately but they are controlled independently in a closed-loop manner.

Vision sensors have been extensively used in robotics (Caron et al. 2013; Chaumette and Hutchinson 2006; Chen et al. 2009; Chesi and Hashimoto 2010; Jara et al. 2014; Nammoto et al. 2013), because of the versatility, low cost and rich information they provide. A main advantage of calibrated camera-based visual servoing is that classical control theory can be directly applied. The drawback is that they depend critically on the accuracy of the calibration. A failure can occur because of the initial calibration errors, aging of components, changes of environmental conditions, maintenance work, etc. Uncalibrated visual servoing approaches are usually simple, computationally efficient and accurate, but they bring a number of challenges such as local minima, singularities or contorted trajectories in the Cartesian space, as well as different practical issues (Ramirez and Jägersand 2016).

A complex problem often addressed in visual servoing is the estimation of the Jacobian matrix in uncalibrated setups (Liang et al. 2015; Piepmeier et al. 2004; Shademan and Jägersand 2012). In Malis (2004) a visual servoing scheme which is invariant to changes in camera-intrinsic parameters is presented. The basic idea in Malis (2004) is to use projective invariance to build the control task from only measured image features. The idea of designing a mapping from the control task to a different space than the image has been used profusely (Adachi and Sato 2004; Cai et al. 2016; Li et al. 2014). Here, a novelty of the approach presented in this paper resides in the control law defined directly in terms of image information and the procedure to show the stability analysis, which is a difficult issue in image-based approaches.

An important limitation in uncalibrated visual servoing arises when the initial pose discrepancy is large and the feature points may leave the camera's field of view. An example solving this limitation by planning trajectories in the image space using uncalibrated stereo cameras is presented in Park and Chung (2003). Moreover, standard uncalibrated visual servoing methods cannot be applied directly to the problem considered here because of the required constrained motions. Therefore, specific control schemes, as the one we propose, need to be designed taking into account the different limitations of the actuators.

Recently, in López-Nicolás et al. (2015) we proposed a vision-based approach to push an object to a target pose with two mobile robots. This approach unifies all the advantages of previous related works such as being sensor-based (vision), closed-loop, stability proven, applicable to non-polygonal objects, performing smooth, continuous and efficient trajectories, etc., and beyond all these it is an uncalibrated approach. This approach was validated by simulation. However, that approach required a rectification step of the input images to remove the projective distortion and recover similarity properties in order to prove the stability of the control scheme.

Here, in this paper, we improve the solution given in López-Nicolás et al. (2015) (1) by eliminating this rectification step of the input images and thus making it fully image-based, (2) by proposing a method for the selection of the pushing points on the object, (3) by adapting it for both mobile robots and manipulators (e.g., a dual-arm robotic platform, see Fig. 1), (4) by proving stability of the proposed solution, and (5) by validating it experimentally. In the pushing task, we considered that each pusher robot interacts with the object through frictional point contacts under the quasi-static motion assumption, which is usually assumed

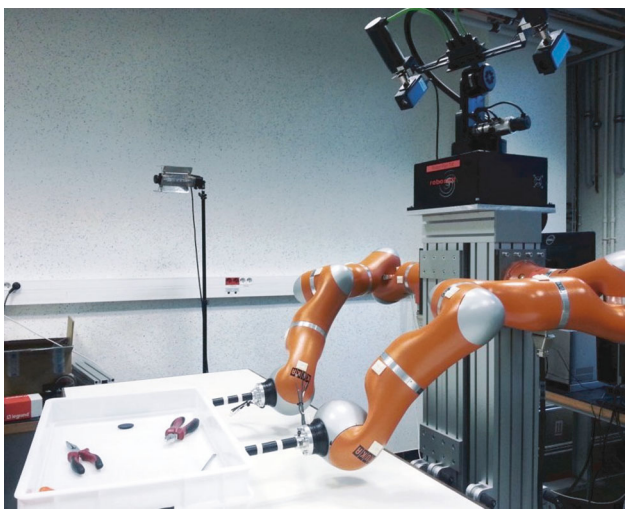

Fig. 1 Kuka dual-arm manipulator platform pushing a box shaped object. One of the top cameras observing the scene is used in an uncalibrated eye-to-hand configuration for the control

in the literature (Lynch 1992), although there are works that take into account the dynamics of the object during pushing (Kopicki et al. 2016; Rezzoug and Gorce 1999). We applied the pushing force with an angle relative to the surface so that it was inside the friction cone and no slip occurred during the pushing if some conditions are met (Gandolfo et al. 1991; Okawa and Yokoyama 1992). This allowed us to be able to push the object like a non-holonomic vehicle actuated only for forward motion.

Consequently we exploited the well-studied contact physics and the non-holonomic vehicle kinematics theories to model and control the pushing task. Moreover, we performed the pushing task only with the image information obtained from an uncalibrated camera looking at the workspace from an arbitrary pose. Here, we consider a conventional camera that can be modeled as a pinhole camera. This allows us to avoid both intrinsic and extrinsic calibration of the camera. The pinhole camera model cannot be assumed if, for example, there is an important image distortion due to a wide angle lens. In that case, an initial calibration of the intrinsic camera parameters to unwarp the distortion would be needed. This methodology simplified the pushing task from a practical point of view and increased the system robustness. We made the following assumptions as in Lynch (1992) for the pushing task:

- All motions and forces are on an obstacle-free plane which is normal to the gravity vector.
- Frictional forces conform to Coulomb's Law (Lynch et al. 1992).
- The friction is uniform between the object and the support plane. This means that the center of friction of the pushed object is the point on the support plane beneath the center of mass of the pushed object (MacMillan 1936).
- Motions are slow enough that inertial forces are negligible, i.e., quasi-static motion assumption. This means that the pushing forces and the frictional forces balance each other.

Shortly, the proposed solution contributes on the existing literature in three ways: (1) It performs pushing manipulation with an uncalibrated image-based visual servoing scheme. This does not need any metric information, calibration, nor rectification. (2) It yields smooth, continuous, and efficient trajectories by imposing non-holonomic velocity constraints on the pushed object. (3) Finally, it proves the local and global stability of the uncalibrated image-based control law.

The rest of this paper goes on as follows: Sect. 2 explains how to actuate an object by pushing with two frictional point contacts; Sect. 3 develops the control law based on non-holonomic motion kinematics for an object with forward motion constraints; Sect. 4 discusses how to compute an optimal motion direction for the pushed object; Sect. 5 proposes an algorithm to find two pushing points on the object; Sect. 6 presents the experimental results; and finally Sect. 7 concludes the paper.

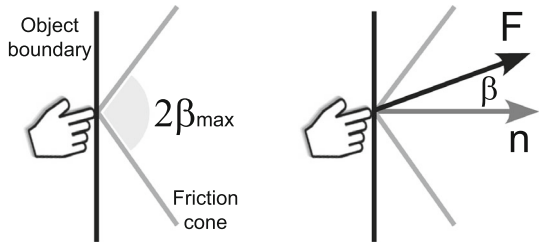

Fig. 2 (Left) Point contact on the object boundary with Coulomb friction μ . (Right) An applied force **F** pointing inside the friction cone

2 Pushing with two frictional point contacts

Here first, we explain the physics of a frictional point contact, then we show how to actuate an object by pushing with two frictional point contacts like a non-holonomic vehicle.

We choose a point contact with friction when friction exists between the fingertip and the object. In a point contact with Coulomb friction model, we can apply a force **F** to an object in any direction as long as it is oriented within the friction-cone. The apex of the friction-cone coincides with the contact point, and the cone axis is aligned with the inward object boundary normal **n**. The cone aperture is defined by the static friction coefficient $\mu = \tan\ \beta_{max} > 0$, see Fig. 2.

Exploiting the above frictional point contact physics, we now would like to move an object like a non-holonomic vehicle by pushing as it is illustrated in Fig. 3. In order to do so, we should push the object from points located on the left and right sides of its mass center **G**, and each contact friction cone should allow to generate a force oriented towards the desired motion direction of the object. Afterwards, we know that from Newton's second law, if the mass of the object is constant, then the linear and angular acceleration of the object are directly proportional to the global force and torque acting on the object and they are in the direction of this force and torque. Thus, up to certain conditions and limitations that will be studied in the following sections, it is possible to move the object like a non-holonomic vehicle by pushing.

This approach is considerably different than other classical solutions such as the one described in Lynch and Mason (1996), Mason (2001). In Mason (2001), a single line contact is used to push the object along a previously planned trajectory in an open-loop manner. That trajectory is defined through a method for finding the instantaneous centers of rotation that will produce stable pushes. Failing to accurately follow this trajectory or the presence of unaccounted perturbations might yield slip or rolling problems. In our proposal,

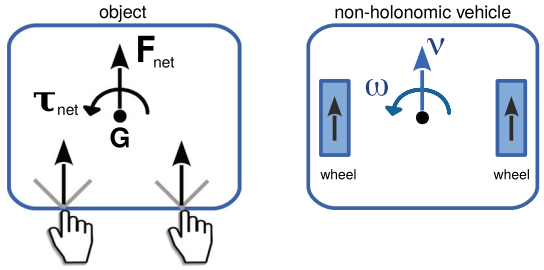

Fig. 3 Moving an object like a non-holonomic vehicle by pushing

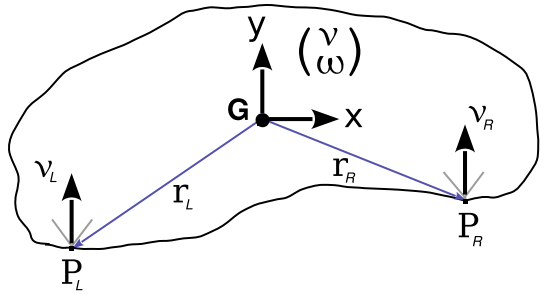

Fig. 4 Geometry of the object with two pushing points. This can be considered as a pseudo non-holonomic vehicle with only forward motion

we use two robots which push the object coordinately but are controlled independently in a closed-loop manner. As we explain in the next sections, we first constrain the control commands with appropriate conditions on the control gains to only produce forward motions. Then, each robot pushes in the selected contact point for which possible slip and rolling are compensated by the image-based closed-loop control.

We remark that the possible complex shape of the object forms a non-symmetric non-holonomic vehicle geometry (see Fig. 4), and unlike a real non-holonomic vehicle, the object can move only forward (and not backward). Given the desired non-holonomic velocity vector (v, ω) of a non-symmetric object, we can compute the velocities of the pushing points as follows:

$$\mathbf{P}_L = \mathbf{G} + \mathbf{r}_L \tag{1}$$
$$\mathbf{v}_L = \dot{\mathbf{G}} + \mathbf{w} \times \mathbf{r}_L \tag{2}$$
$$\mathbf{P}_R = \mathbf{G} + \mathbf{r}_R \tag{3}$$
$$\mathbf{v}_R = \dot{\mathbf{G}} + \mathbf{w} \times \mathbf{r}_R \tag{4}$$

where $\mathbf{r}_L = [r_{Lx}, r_{Ly}, 0]^T$ and $\mathbf{r}_R = [r_{Rx}, r_{Ry}, 0]^T$ are the vectors from the mass center to the left and right pushing points $\mathbf{P}_L$ and $\mathbf{P}_R$; $\mathbf{v}_L$ and $\mathbf{v}_R$ are the velocities of the pushing points; $\dot{\mathbf{G}} = v\ \mathbf{y}$ is the velocity of the mass center with the unit vector $\mathbf{y} = [0, 1, 0]^T$ of the fixed motion direction and the linear speed v; and $\mathbf{w} = \omega\ \mathbf{z}$ is the rotational velocity vector with the rotational axis $\mathbf{z} = [0, 0, 1]^T$ and the angular speed ω. We can simplify and rewrite the velocities of the pushing points for 2D planar space from (2) and (4) as below:

$$\mathbf{v}_L = \begin{bmatrix} 0 & -r_{Ly} \\ 1 & r_{Lx} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$
$$\mathbf{v}_R = \begin{bmatrix} 0 & -r_{Ry} \\ 1 & r_{Rx} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{5}$$

3 Forward non-holonomic motion

In this section, we develop the kinematic model, the control law, and the necessary conditions of an object which performs only forward non-holonomic motion.

3.1 Kinematics

Let us define an object frame fixed to the object and defined by $\mathbf{y}$ and $\mathbf{G}$ (See Fig. 5). We also denote the object frame in the desired position, fixed with respect to the world frame, with $\mathbf{y}^*$ and $\mathbf{G}^*$. Let the state vector of the object in polar coordinates, between the current and target locations, be $\mathbf{x} = [\,\rho,\ \alpha,\ \phi\,]^T$ in the real world plane π_W (see Figs. 5, 6). Here ρ is the distance of the object to the desired position, ϕ is the alignment error of the object with respect to the desired orientation, and α is the alignment error of the fixed motion direction of the object towards the desired position. The object state vector is thus defined as a relative alignment error with respect to the target configuration:

$$\rho = \| \mathbf{G}^* - \mathbf{G} \|$$
$$\alpha = \text{atan2}\,(\ (\mathbf{u} \times \mathbf{y})_z\ ,\ \mathbf{u}^T\ \mathbf{y}\ )$$
$$\phi = \text{atan2}\,(\ (\mathbf{y}^* \times \mathbf{y})_z\ ,\ \mathbf{y}^{*T}\ \mathbf{y}\ ) \tag{6}$$

where $\mathbf{u} = (\ \mathbf{G}^* - \mathbf{G}\ )\ /\ \rho$, and where $(\cdot)_z$ represents the z-axis component of an associated vector. Afterwards, let the

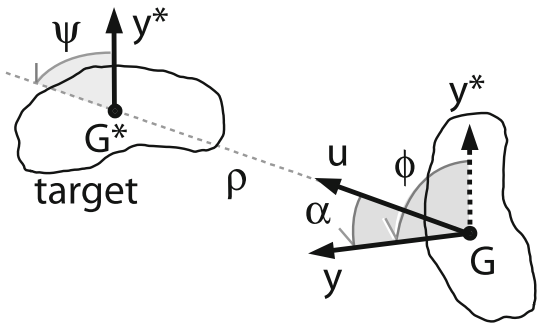

Fig. 5 State variables of the object with respect to a target pose. The object frame attached to the object is defined by $\mathbf{y}$ and $\mathbf{G}$. The same frame with the object in the target location is denoted with $\mathbf{y}^*$ and $\mathbf{G}^*$, being this target frame fixed in the world frame. The speed v is in the direction of $\mathbf{y}$ and ω is positive counter-clockwise

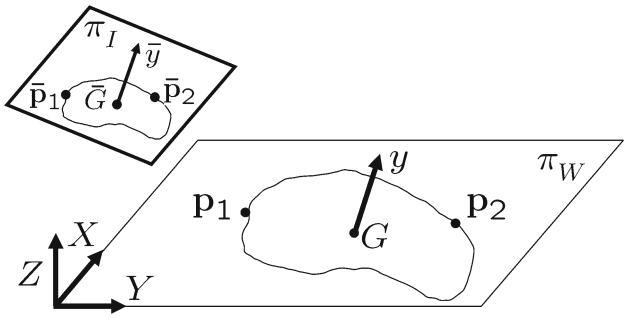

Fig. 6 World plane showing its coordinate frame. The image plane of the camera aiming at the scene is represented on top left. The map between points in the world plane π_w and their image in π_I is a homography

desired motion of the object be the linear speed v and the angular speed ω, then the non-holonomic kinematics of the object can be written as follows (López-Nicolás and Sagüés 2011):

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \frac{1}{\rho}\sin\alpha & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{7}$$

This previous kinematics will be used to model the object when pushed. However, note that the idea is not to assume that the system behaves under this model but to impose, using adequate constraints presented in the following sections, this behavior to the system.

3.2 Relating image plane to Euclidean plane

We use a conventional uncalibrated camera looking at the workspace. This camera is assumed to comply with the pinhole model. Then, neither intrinsic nor extrinsic calibration is required. The input of the control law consists of the images acquired with the camera and a prior target image with the object in the desired pose. Next we analyze the relation of the state variables between the images and the real world. The goal is to show that the state variables computed from the image plane are related to the corresponding variables in the real world with a bounded matrix $\mathbf{T}$. This property is used in the subsequent analysis to establish the stability conditions on the control scheme.

First, let us formulate the following assumption regarding the system configuration: The relative position between the camera and the real world plane where the motion of the object occurs is not in a degenerate configuration. In particular, the camera center does not lie in the world plane. More specifically, we assume that the camera is above the world plane pointing to the workspace where the task is performed.

Proposition 1 *The relation between the measured state variables computed from the image plane π_I and the physical world plane π_W can be written down as $\bar{\mathbf{x}} = \mathbf{T}\,\mathbf{x}$:*

$$\underbrace{\begin{bmatrix} \bar{\rho} \\ \bar{\alpha} \\ \bar{\phi} \end{bmatrix}}_{\bar{\mathbf{x}}} = \underbrace{\begin{bmatrix} T_\rho & 0 & 0 \\ 0 & T_\alpha & 0 \\ T_0 & 0 & T_\phi \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} \rho \\ \alpha \\ \phi \end{bmatrix}}_{\mathbf{x}} \tag{8}$$

where $\bar{\mathbf{x}} \in \pi_I$ is the vector of measured state variables from the input image, and $T_\rho(\mathbf{x})$, $T_\alpha(\mathbf{x})$, $T_\phi(\mathbf{x})$ and $T_0(\mathbf{x})$ are scalar functions with lower and upper bounds as follows:

$$0 < T_\rho^{min} \leq T_\rho \leq T_\rho^{max} \tag{9}$$
$$0 < T_\alpha^{min} \leq T_\alpha \leq T_\alpha^{max} \tag{10}$$
$$0 < T_\phi^{min} \leq T_\phi \leq T_\phi^{max} \tag{11}$$
$$T_0^{min} \leq T_0 \leq T_0^{max} \tag{12}$$

Proof Let us define two points $(\mathbf{p}_1, \mathbf{p}_2)$ on the physical world plane π_W and $(\bar{\mathbf{p}}_1, \bar{\mathbf{p}}_2)$ their image projections. We choose the world coordinate frame with Z-axis orthogonal to the physical world plane and origin in this plane as shown in Fig. 6. Then, the mapping between the points on the X-Y world plane and the image plane is a homography $\mathbf{H}$ such as

$$\bar{\mathbf{p}}_1 = \lambda_1 \mathbf{H}\,\mathbf{p}_1 \tag{13}$$
$$\bar{\mathbf{p}}_2 = \lambda_2 \mathbf{H}\,\mathbf{p}_2 \tag{14}$$

with finite scalar coefficients $\lambda_1, \lambda_2 \neq 0$. We also define $\mathbf{p} = \mathbf{p}_2 - \mathbf{p}_1$, $\bar{\mathbf{p}} = \bar{\mathbf{p}}_2 - \bar{\mathbf{p}}_1$. By definition we have $\bar{\rho} = ||\bar{\mathbf{p}}||$ and $\rho = ||\mathbf{p}||$. Then, we can write function T_ρ as follows

$$T_\rho = \frac{||\bar{\mathbf{p}}||}{||\mathbf{p}||} = \frac{||\mathbf{H}\,(\lambda_2\,\mathbf{p}_2 - \lambda_1\,\mathbf{p}_1)||}{||\mathbf{p}_2 - \mathbf{p}_1||}\,. \tag{15}$$

Without loss of generality we can choose $\mathbf{p}_2 = \mathbf{0}$, and then

$$T_\rho = \frac{||\lambda_1\,\mathbf{H}\,\mathbf{p}_1||}{||\mathbf{p}_1||} = \frac{|\lambda_1|\;||\mathbf{H}\,\mathbf{p}_1||}{||\mathbf{p}_1||}\,. \tag{16}$$

Denoting $\lambda_{max}(\mathbf{H})$ and $\lambda_{min}(\mathbf{H})$ the maximum and minimum eigenvalues of the homography matrix $\mathbf{H}$ we can write:

$$T_\rho^{min} = |\lambda_1|\,\lambda_{min}(\mathbf{H}) \leq T_\rho \leq |\lambda_1|\,\lambda_{max}(\mathbf{H}) = T_\rho^{max}\,. \tag{17}$$

Note that if $\mathbf{p}_1 = \mathbf{p}_2$ we have $\rho = 0 \Leftrightarrow \bar{\rho} = 0$, otherwise $\rho > 0 \Leftrightarrow \bar{\rho} > 0$. In the former singular case, taking limits to T_ρ produces $|\lambda_1|\;||\mathbf{H}||$. Therefore, bounds of T_ρ in (9) can always be defined as in Eq. (17).

Regarding $T_\alpha = \bar{\alpha}/\alpha$, note that a planar projective transformation such as the homography $\mathbf{H}$ is known to preserve collinearity. Therefore, $\bar{\alpha} = 0 \Leftrightarrow \alpha = 0$ (or alternatively $\bar{\alpha} = \pm\pi \Leftrightarrow \alpha = \pm\pi$). As a consequence we have that the angle between two intersecting lines in the real world

and the projection of this angle to the image plane have the same sign, i.e. $\alpha \in (0,\pi) \Leftrightarrow \bar{\alpha} \in (0,\pi)$ or else $\alpha \in (-\pi,0) \Leftrightarrow \bar{\alpha} \in (-\pi,0)$. We can write function T_α as follows

$$T_\alpha = \frac{\bar{\alpha}}{\alpha} = \frac{\text{atan2}((\bar{\mathbf{u}} \times \bar{\mathbf{y}})_z\,,\, \bar{\mathbf{u}}^T\,\bar{\mathbf{y}})}{\text{atan2}((\mathbf{u} \times \mathbf{y})_z\,,\, \mathbf{u}^T\,\mathbf{y})}\,, \tag{18}$$

which is clearly positive and bounded for any $\alpha \neq 0$ (i.e. $\mathbf{u}$ and $\mathbf{y}$ not collinear). In the singular case of $\mathbf{u}$ and $\mathbf{y}$ collinear, we have small values for α and $\bar{\alpha}$. Therefore we consider $\alpha \simeq \tan\alpha$ and $\bar{\alpha} \simeq \tan\bar{\alpha}$. Since α and $\bar{\alpha}$ have same sign we can write

$$\begin{aligned} T_\alpha &= \frac{|\bar{\alpha}|}{|\alpha|} \simeq \frac{\|\bar{\mathbf{u}} \times \bar{\mathbf{y}}\|\ \|\mathbf{u}^T\,\mathbf{y}\|}{\|\mathbf{u} \times \mathbf{y}\|\ \|\bar{\mathbf{u}}^T\,\bar{\mathbf{y}}\|} \\ &= \frac{\|\lambda_u\,\mathbf{H}\,\mathbf{u} \times \lambda_y\,\mathbf{H}\,\mathbf{y}\|\ \|\mathbf{u}^T\,\mathbf{y}\|}{\|\mathbf{u} \times \mathbf{y}\|\ \|\lambda_u\,\mathbf{u}^T\,\mathbf{H}^T\,\mathbf{H}\,\mathbf{y}\,\lambda_y\|} \\ &= \frac{\|\mathbf{H}\,\mathbf{u} \times \mathbf{H}\,\mathbf{y}\|\ \|\mathbf{u}^T\,\mathbf{y}\|}{\|\mathbf{u} \times \mathbf{y}\|\|\mathbf{u}^T\mathbf{H}^T\mathbf{H}\mathbf{y}\|} = \frac{|\mathbf{H}|\|\mathbf{H}^{-T}\,(\mathbf{u} \times \mathbf{y})\|\|\mathbf{u}^T\mathbf{y}\|}{\|\mathbf{u} \times \mathbf{y}\|\ \|\mathbf{u}^T\,\mathbf{H}^T\,\mathbf{H}\,\mathbf{y}\|}, \end{aligned} \tag{19}$$

with finite scalar coefficients $\lambda_u, \lambda_y \neq 0$ and $|\mathbf{H}|$ the determinant of matrix $\mathbf{H}$. Properties of quadratic forms lead to

$$\begin{aligned} \frac{|\mathbf{H}|\lambda_{min}(\mathbf{H}^{-T})}{\lambda^2_{max}(\mathbf{H})} &\leq \frac{|\mathbf{H}|\|\mathbf{H}^{-T}\,(\mathbf{u} \times \mathbf{y})\|\|\mathbf{u}^T\,\mathbf{y}\|}{\|\mathbf{u} \times \mathbf{y}\|\ \|\mathbf{u}^T\mathbf{H}^T\mathbf{H}\,\mathbf{y}\|} \\ &\leq \frac{|\mathbf{H}|\lambda_{max}(\mathbf{H}^{-T})}{\lambda^2_{min}(\mathbf{H})}. \end{aligned} \tag{20}$$

Therefore, bounds of T_α in (10) can always be defined as in Eq. (20).

The case of $(\phi, \bar{\phi})$ in (8) is different than $(\alpha, \bar{\alpha})$ because $\phi = 0$ (or $\pm\pi$) does not imply $\bar{\phi} = 0$ (or $\pm\pi$). Therefore $\phi = 0$ and $\bar{\phi} \neq 0$ is a singular case. It appears when two lines are parallel ($\mathbf{y} \times \mathbf{y}^* = \mathbf{0}$) but not collinear ($\mathbf{u} \times \mathbf{y}^* \neq \mathbf{0}$) in the world plane. This is a consequence of perspective projection which makes parallel lines intersect in the image. In (8), multiplying ϕ, when $\phi = 0$, only with a positive constant T_ϕ cannot produce $\bar{\phi} \neq 0$. We handle this by adding some contribution from ρ exploiting that in the singular case we have $\rho \neq 0$ (since $\phi = 0$ and $\rho = 0$ implies $\bar{\phi} = 0$):

$$\bar{\phi} = T_0\,\rho + T_\phi\,\phi\,, \tag{21}$$

where we define functions T_ϕ and T_0 as follows

$$T_\phi = \begin{cases} \frac{\text{atan2}((\bar{\mathbf{y}}^* \times \bar{\mathbf{y}})_z\,,\bar{\mathbf{y}}^{*T}\,\bar{\mathbf{y}})}{\text{atan2}((\mathbf{y}^* \times \mathbf{y})_z\,,\mathbf{y}^{*T}\,\mathbf{y})}\,, & if\ (\phi \neq 0) \\ \frac{|\mathbf{H}|\ \|\mathbf{H}^{-T}\|}{\lambda^2_{min}(\mathbf{H})}\,, & if\ (\phi = 0) \end{cases} \tag{22}$$

$$T_0 = \begin{cases} \text{atan2}((\bar{\mathbf{y}}^* \times \bar{\mathbf{y}})_z\,,\bar{\mathbf{y}}^{*T}\,\bar{\mathbf{y}})/\rho\,, & if\ (\phi = 0) \wedge (\bar{\phi} \neq 0) \\ 0\,, \quad if\ (\phi \neq 0) \vee ((\phi = 0) \wedge (\bar{\phi} = 0)) \end{cases} \tag{23}$$

By defining arbitrarily small thresholds $\phi_0, \bar{\phi}_0 \in \Re^+$ we can obtain the following bounds: $T^{max}_\phi = \pi\,/\,\phi_0$ and $T^{min}_\phi = \bar{\phi}_0\,/\,\pi$. For the singular case, the definitions of T_ϕ and T_0 given in (22) and (23) allow to express correctly the mapping between ϕ and $\bar{\phi}$. For the non-singular case, ϕ behaves similar to α, therefore it is bounded. Then, bounds for T_ϕ and T_0 can be found from (22) and (23). □

3.3 Control law

In this section we describe the proposed control law, which is inspired from López-Nicolás and Sagüés (2011). In López-Nicolás and Sagüés (2011), the position-based control scheme is based on Euclidean space information, and it is applied to a non-holonomic mobile robot equipped with a calibrated onboard camera. Here, we adapt this control law to be computed from uncalibrated image information, and then apply it to a system where an object is pushed on a plane and observed by an external uncalibrated camera looking at the scene. We define the object's desired motion with this new control law for this new system (i.e., object manipulation by pushing) from the computed state variables as follows:

$$v = k_\rho\ \bar{\rho}\ \cos\bar{\alpha} \tag{24}$$

$$\omega = k_\phi\ \bar{\phi}\ -\ k_\alpha\ \bar{\alpha} \tag{25}$$

where k_ρ, k_α, and k_ϕ are positive constant control gains.

Theorem 1 (Local exponential stability) *If the chosen control gains satisfy the following condition:*

$$T^{min}_\alpha\,k_\alpha\ -\ (T^{max}_\phi\,k_\phi\ +\ T^{max}_\rho\,k_\rho)\ >\ 0 \tag{26}$$

then the control law (24-25) *is locally exponentially stable.*

Proof Using (8), (24) and (25) in (7), we can obtain the closed-loop system described as below:

$$\begin{aligned} \dot{\rho} &= -k_\rho\,T_\rho\,\rho\,\cos(T_\alpha\,\alpha)\,\cos\alpha \\ \dot{\alpha} &= k_\phi\,T_\phi\,\phi + k_\phi\,T_0\,\rho - k_\alpha\,T_\alpha\,\alpha + k_\rho\,T_\rho\,\cos(T_\alpha\,\alpha)\,\sin\alpha \\ \dot{\phi} &= k_\phi\,T_\phi\,\phi + k_\phi\,T_0\,\rho - k_\alpha\,T_\alpha\,\alpha \end{aligned} \tag{27}$$

The first-order Taylor series approximation of this closed-loop system (27) around the equilibrium state ($\mathbf{x} = \mathbf{0}$) gives the following linearized system $\dot{\mathbf{x}} \approx \mathbf{A}\,\mathbf{x}$:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\phi} \end{bmatrix} \approx \underbrace{\begin{bmatrix} -k_\rho T_\rho & 0 & 0 \\ k_\phi T_0 & k_\rho T_\rho - k_\alpha T_\alpha & k_\phi\,T_\phi \\ k_\phi T_0 & -k_\alpha\,T_\alpha & k_\phi\,T_\phi \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \rho \\ \alpha \\ \phi \end{bmatrix} \tag{28}$$

This system is locally exponentially stable if and only if the eigenvalues of the linearized system matrix are negative. This

yields to the following condition:

$$T_\alpha\, k_\alpha \;-\; T_\phi\, k_\phi \;-\; T_\rho\, k_\rho \;>\; 0 \tag{29}$$

The worst case is given then by (26), and satisfying (26) is sufficient for local exponential stability. □

Theorem 2 (Global asymptotical stability) *The control law* (24-25)*, with gains satisfying* (26)*, is globally asymptotically stable if the following condition also holds:*

$$1 > 4\, k_\rho\, T_\rho^{max}\, \lambda_{max}(\mathbf{P})\ , \tag{30}$$

being $\lambda_{max}(\mathbf{P})$ *the maximum eigenvalue of* $\mathbf{P}$*, where* $\mathbf{P}$ *is:*

$$\begin{aligned}
\mathbf{P} \;&=\; \frac{1}{2\, T_\phi\, k_\phi\, T_\rho\, k_\rho\, N} \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \\
&with \quad N = (T_\alpha k_\alpha - T_\phi k_\phi - T_\rho k_\rho)\ , \qquad (31) \\
p_{11} \;&=\; \left((T_0 k_\phi)^2 (T_\alpha k_\alpha + T_\phi k_\phi) + T_\alpha k_\alpha T_\phi k_\phi N\right) / (T_\alpha k_\alpha) \\
p_{12} = p_{21} \;&=\; T_0 k_\phi \left(T_\phi k_\phi T_\rho k_\rho + (T_\alpha k_\alpha - T_\phi k_\phi)^2\right) / (T_\alpha k_\alpha) \\
p_{13} = p_{31} \;&=\; T_0 k_\phi \left(T_\alpha k_\alpha T_\rho k_\rho - (T_\alpha k_\alpha - T_\phi k_\phi)^2\right) / (T_\alpha k_\alpha) \\
p_{22} \;&=\; (T_\alpha k_\alpha)^2 + (T_\phi k_\phi)^2 + T_\phi k_\phi T_\rho k_\rho \\
p_{23} = p_{32} \;&=\; -(T_\alpha k_\alpha)^2 - (T_\phi k_\phi)^2 + T_\alpha k_\alpha T_\rho k_\rho \\
p_{33} \;&=\; (T_\alpha k_\alpha - T_\rho k_\rho)^2 + (T_\phi k_\phi)^2 + T_\phi k_\phi T_\rho k_\rho \qquad (32)
\end{aligned}$$

Proof We define the candidate Lyapunov function as:

$$V(\mathbf{x}) \;=\; \mathbf{x}^T\ \mathbf{P}\ \mathbf{x} \tag{33}$$

where $\mathbf{P}$ is a symmetric positive definite matrix to be found. We write the system dynamics (27) in the following form:

$$\dot{\mathbf{x}} \;=\; \mathbf{A}\,\mathbf{x} \;+\; \mathbf{f}(\mathbf{x}) \tag{34}$$

with $\mathbf{A}$ defined in (28) and $\mathbf{f}(\mathbf{x})$ as below:

$$\mathbf{f}(\mathbf{x}) \;=\; k_\rho\, T_\rho \begin{bmatrix} \rho \;-\rho\, \cos\alpha\ \cos(T_\alpha\, \alpha) \\ \sin\alpha\ \cos(T_\alpha\, \alpha) - \alpha \\ 0 \end{bmatrix} \tag{35}$$

Notice that the matrix $\mathbf{A}$ is Hurwitz given that constraint (26) holds. The derivative of (33) yields:

$$\begin{aligned}
\dot{V} &= \dot{\mathbf{x}}^T\, \mathbf{P}\, \mathbf{x} \;+\; \mathbf{x}^T\, \mathbf{P}\, \dot{\mathbf{x}} \\
&= \mathbf{x}^T\, (\mathbf{A}^T\, \mathbf{P} \;+\; \mathbf{P}\, \mathbf{A})\, \mathbf{x} \;+\; \mathbf{f}^T\, \mathbf{P}\, \mathbf{x} \;+\; \mathbf{x}^T\, \mathbf{P}\, \mathbf{f} \\
&= -\mathbf{x}^T\, \mathbf{Q}\, \mathbf{x} \;+\; 2\, \mathbf{x}^T\, \mathbf{P}\, \mathbf{f} \qquad (36)
\end{aligned}$$

where $\mathbf{A}^T\, \mathbf{P} \;+\; \mathbf{P}\, \mathbf{A} \;=\; -\mathbf{Q}$ and $\mathbf{Q} \;=\; \mathbf{Q}^T \;>\; 0$. Choosing $\mathbf{Q} = \mathbf{I}$ and solving for $\mathbf{P} = \mathbf{P}^T$, we obtain $\mathbf{P}$ as defined in (31). It can be shown that $\mathbf{P}$ is positive definite if constraint (29) holds by checking that all leading principal minors of $\mathbf{P}$ are positive. Now, we work out (36) with the goal of showing that $\dot{V} \;<\; 0$. It can be demonstrated that, there exists a constant $M \in \Re$ in such a way that:

$$||\mathbf{f}(\mathbf{x})|| \;<\; M\, ||\mathbf{x}||\ . \tag{37}$$

In particular, the norms of the vectors involved in the inequality (37) results in:

$$\begin{aligned}
k_\rho^2\, T_\rho^2 \left(\rho^2 (1 - \cos\alpha\ \cos(T_\alpha\, \alpha))^2 + (\sin\alpha\ \cos(T_\alpha\, \alpha) - \alpha)^2\right) \\
\leq\ (2\, k_\rho\, T_\rho\, \rho)^2 + (2\, \alpha)^2 \;<\; M^2\, (\rho^2 + \alpha^2 + \phi^2) \qquad (38)
\end{aligned}$$

which is true if

$$M^2 \;>\; (2\, k_\rho\, T_\rho)^2 \tag{39}$$

Therefore, we choose $M^{max} \;>\; (2\, k_\rho\, T_\rho^{max})$ in such a way that (37) holds. Then, we can develop (36) as follows:

$$\begin{aligned}
\dot{V} &\leq -\mathbf{x}^T\, \mathbf{Q}\, \mathbf{x} \;+\; 2\, ||\mathbf{P}\, \mathbf{x}||\, M\, ||\mathbf{x}|| \\
&\leq -\lambda_{min}(\mathbf{Q}) ||\mathbf{x}||^2 \;+\; 2\, M\, \lambda_{max}(\mathbf{P}) ||\mathbf{x}||^2 \\
&= -\left(\lambda_{min}(\mathbf{Q}) \;-\; 2\, M \lambda_{max}(\mathbf{P})\right)\, ||\mathbf{x}||^2 \qquad (40)
\end{aligned}$$

Therefore, $\dot{V} \;<\; 0$ if the next inequality holds:

$$\lambda_{min}(\mathbf{Q}) \;>\; 2\, M^{max}\, \lambda_{max}(\mathbf{P}) \tag{41}$$

where $\lambda_{min}(\mathbf{Q}) = 1$, which yields (30). Therefore the control law is globally asymptotically stable if (26) and (30) hold. □

Finally, the input velocities of the pushers $\mathbf{v}_L$ and $\mathbf{v}_R$ from the desired motion of the object—the speeds v and ω in (24) and (25)—are computed using (5).

The condition of local stability (29) can be interpreted intuitively by noticing that it imposes that $T_\alpha\, k_\alpha$ must be greater than the sum of $T_\phi\, k_\phi$ and $T_\rho\, k_\rho$. This means that the convergence of α must be faster than ρ and ϕ ensuring alignment of the system with the desired direction of motion and following it. This avoids undesirable behaviors such as spiral trajectories around the target position. Global stability also takes into account the non-linear term $\mathbf{f}(\mathbf{x})$ with condition (30), which guarantees that non-linear terms (35) are bounded by the convergence rate of $\mathbf{A}$ (34), so the system copes with non-holonomic motion constraints guaranteeing global convergence.

Remark 1 We would like to remark that these speeds are normally for non-holonomic vehicles and they can move the vehicles forward and backward. In case of a pushed object, we should avoid backward motion since we cannot pull. Next subsection emphasizes the required conditions for forward motion maneuvers only.

3.4 Forward motion conditions

The first condition that should hold for the state variables to guarantee the forward motion ($v \geq 0$) is as below:

$$|\bar{\alpha}| \leq \pi/2 \quad \text{and} \quad |\bar{\psi}| \leq \pi/2 \tag{42}$$

where $\bar{\psi}$ is the image projection of ψ (see Fig. 5) and can be defined as $\bar{\psi} = \bar{\phi} - \bar{\alpha}$. Constraint (42) avoids cusps in the trajectory of the object which cause backward motion. This constraint is guaranteed by the proper choice of the motion direction $\mathbf{y}$ of the object. Next section explains how to compute the motion direction of the object.

The second condition that we should also impose while pushing the object is to limit the angular speed of the object:

$$|\omega| \leq v / r \tag{43}$$

where r is the distance between a pushing point and the mass center of the object (see Fig. 4). This distance r is computed as follows:

$$r = \begin{cases} \|\mathbf{r}_L\| & if\ \omega > 0 \\ \|\mathbf{r}_R\| & if\ \omega < 0 \\ \infty & if\ \omega = 0 \end{cases} \tag{44}$$

Constraint (43) avoids velocities of the pushing points $\mathbf{v}_L$ and $\mathbf{v}_R$ to have negative values for the motion axis $\mathbf{y}$ (5). This prevents the control scheme from asking for backward motions by constraining the rotational motion speed ω with respect to the forward motion speed v. Not imposing this constraint will result in non-viable motions. Another complex issue to deal with is to guarantee the non-slip condition and continuous contact of the pushers with the object, since constraint (43) is a necessary condition but not sufficient. Thus, in our approach each robot pushes in the selected contact point for which possible slip and rolling are compensated by the image-based closed-loop control. The procedure to deal with this issue is presented in Sect. 6.1 for the case of mobile robots and in Sect. 6.2 for the case of the dual arm manipulator.

The previous condition (43) is required to limit the angular speed of the object. In order to guarantee that the control commands do not break this condition, we seek for the relations of the control gains that impose this constraint. To do so, next proposition elaborates on (43) using the proposed control law (24) and (25).

Proposition 2 *Constraint* $|\omega| \leq v/r$ (43) *is satisfied if the following two conditions hold:*

$$\frac{\pi}{2} \left| 2\,k_\phi - k_\alpha \right| \leq \varepsilon_\alpha \left(\frac{T_\rho^{min}\ k_\rho\ \bar{\rho}_+}{\bar{r}} - k_\alpha \right) \tag{45}$$

$$2 \left(k_\rho\, T_\rho^{min} + k_\phi\, T_\phi^{min} \right) > k_\alpha\, T_\alpha^{max} \tag{46}$$

where $\bar{\rho}_+ > 0$*; where* $\bar{r} = T_\rho\, r$ *is the distance between the mass center and the corresponding pushing point of the object in the plane* π_I *with* T_ρ *defined as in* (8)*; and where* ε_α *is an arbitrary small positive angle (hence* $\sin(\varepsilon_\alpha) \approx \varepsilon_\alpha$*).*

Proof Next, we study the conditions when constraint (43) is violated. The worst cases for this constraint can be divided according the value of $\bar{\rho}$ into $\bar{\rho} = \bar{\rho}_+ > 0$ and $\bar{\rho} = 0$.
Case $\bar{\rho} = \bar{\rho}_+ > 0$.

We can rewrite the angular speed constraint $|\omega| \leq v/r$ using (8), (24) and (25) as follows:

$$\left| k_\phi\, \bar{\phi} - k_\alpha\, \bar{\alpha} \right| \leq \frac{T_\rho\ k_\rho\ \bar{\rho}_+ \cos\bar{\alpha}}{\bar{r}}\ , \tag{47}$$

where we consider the case of $\bar{\rho} = \bar{\rho}_+$ any positive value. Then, the worst case which violates the above condition happens for $\bar{\alpha} = \pm\pi/2$, when the motion direction of the object is orthogonal to the direction towards the target. In this case, $v = 0$ and the constraint $|\omega| \leq v/r$ is violated. This is a singular case that can be avoided in practice by, for example, applying the proposed algorithm for motion direction selection (see Sect. 4). We thus set the worst case as $\bar{\alpha} = \pm\pi/2 \mp \varepsilon_\alpha$ where ε_α is an arbitrary small positive angle. Addition of $\pm\varepsilon_\alpha$ to $\bar{\alpha}$ also keeps $v > 0$.

Regarding $\bar{\phi}$, it can be deduced that the worst case corresponds to $\bar{\phi} = \pm\pi$. This happens when the motion direction of the object is oriented backward with respect to desired configuration. Note that, although $\bar{\phi} = \mp\pi$ could seem a worse case, it cannot happen with $\bar{\alpha} = \pm\pi/2$ at the same time given that we express all angles in $[-\pi\ , \pi]$. For instance, if $\bar{\alpha} = \pi/2$ and given that by definition $\bar{\alpha} = \bar{\phi} - \bar{\psi}$, we have that $\bar{\phi} = -\pi$ leads to $\bar{\psi} = -3\pi/2$ whereas the correct expression of the angles would be $\bar{\psi} = \pi/2$ resulting in $\bar{\phi} = \pi$.

Finally, (47) can be rewritten for the worst case values of $\bar{\alpha}$, $\bar{\phi}$ and T_ρ as (45). Note that condition (45) relates gains k_ϕ and k_α, which are directly related with ω, with gain k_ρ, which is related with v. Thus, this condition is constraining ω with respect v to avoid larger rotations at the beginning of the motion.
Case $\bar{\rho} = 0$.

Condition (45) allows tuning the control gains to avoid control law saturation when $v > 0$. Nevertheless, it does not guarantee $|\omega| \leq v/r$ at the end of the motion period, when $\bar{\rho} \to 0$ and therefore $v \to 0$. In that case, we must guarantee that the convergence rate of ω is higher than v. This case appears when the system is getting closer to the equilibrium state. Therefore, we can consider here the first-order Taylor series approximation of the closed-loop system given in (28).

The eigenvalues of this system are as follows:

$$\lambda_\rho \;=\; -k_\rho\, T_\rho\,, \quad \lambda_{\alpha,\phi} \;=\; \frac{-N \mp \sqrt{(N)^2 - 4\,k_\rho\, T_\rho\, k_\phi\, T_\phi}}{2} \tag{48}$$

with N defined in (31). Since the convergence rates of the state variables are given by their eigenvalues:

$$\bar{\rho}(t) \;=\; \bar{\rho}_0\, e^{(\lambda_\rho\, t)} \quad \text{thus} \quad v \;\approx\; k_\rho\, \bar{\rho}_0\, e^{(\lambda_\rho\, t)}\,, \tag{49}$$

we derive the conditions for (48) such that $|\lambda_\rho|$ is the smallest eigenvalue. In particular, we look for condition:

$$|\lambda_\alpha| \;\geqslant\; |\lambda_\phi| \;>\; |\lambda_\rho|\,. \tag{50}$$

We can deduce from (48) and (50) the following expression

$$2\,(k_\rho\, T_\rho \;+\; k_\phi\, T_\phi) \;>\; k_\alpha\, T_\alpha\,, \tag{51}$$

from which the worst case leads to (46). The intuition on (46) is that the pushed object should approximate the desired position with the target orientation. In particular, note that whereas (26) imposes a higher value for k_α over k_ρ and k_ϕ, condition (46) imposes that k_α should be below a combination of k_ρ and k_ϕ. This is to avoid large corrections on the orientation from the control law with low linear speed, which would be impractical due to the angular speed limitation in the pushing of the object.

We can thus sustain the constraint of the angular speed given in (43) by satisfying (45) and (46). □

3.5 Illustrative scenario on bounds and stability

We now illustrate with a numerical example the stability conditions. In this scenario, Proposition 1 and Theorems 1, 2 are used to validate convergence to the solution. The object is placed in position $(x, y, z) = (-500, 1000, 0)$ mm with orientation on the plane of $\phi = 45°$. The target pose is the origin. The camera has a focal length of 6 mm and it is located in position $(-2000, 1000, 2000)$ mm with orientation angles of $(-20°, -10°, 0°)$. The size of the acquired images is 640×640 pixels. The control gains have been set as follows: $k_\rho = 0.2$, $k_\alpha = 0.7$, and $k_\phi = 0.4$. The simulated results using the proposed control law (24–25) give the following bounds in (9–12): $(T_\rho^{min}, T_\rho^{max}) = (0.223, 0.247)$, $(T_\alpha^{min}, T_\alpha^{max}) = (0.885, 1.070)$, $(T_\phi^{min}, T_\phi^{max}) = (0.888, 1.043)$, $(T_0^{min}, T_0^{max}) = (2.546, 5.193)$, and $\lambda_{max}(\mathbf{P}) = 0.733$. It can be checked that the control parameters and these values hold the stability conditions (26) and (30).

4 The direction to push

In this section, we propose a way to assign an optimal motion direction to the object, i.e. the $\mathbf{y}$ axis, such that the maneuvers performed toward the target are minimized. In the measurable state vector $\bar{\mathbf{x}}$, the state variables $\bar{\rho}$ and $\bar{\phi}$ are computed from the given input images of the initial and desired poses of the object. However, $\bar{\alpha}$ can be arbitrarily defined by determining the motion direction (i.e. defining the vector $\mathbf{y}$). Therefore, motion direction $\mathbf{y}$ may be defined from the best value of $\bar{\alpha}$. In order to find the best motion direction, we propose to minimize the Lyapunov function $V(\mathbf{x})$ given in (33) for $\bar{\alpha}$ (see Theorem 2) when the object is in its initial configuration before starting the pushing task. Here, we first rewrite the Lyapunov function in terms of the measurable state vector $\bar{\mathbf{x}}$ using (8) which yields:

$$V(\bar{\mathbf{x}}) \;=\; \bar{\mathbf{x}}^T\, \mathbf{T}^{-T}\, \mathbf{P}\, \mathbf{T}^{-1}\, \bar{\mathbf{x}}\,. \tag{52}$$

We then minimize (52) by subjecting to condition (42):

$$\min_{\bar{\alpha}} V(\bar{\mathbf{x}}) \quad \text{subject to} \quad \bar{\alpha}^2 \;\leq\; \pi^2/4 \quad \text{and} \quad \bar{\psi}^2 \;\leq\; \pi^2/4 \tag{53}$$

so that the solution $\bar{\alpha}$ does not violate forward motion. The values of $\mathbf{T}$ are bounded scalar functions, but they are unknown in practice. In order to compute (53), and since the actual values of $\mathbf{T}$ will not be computed in this uncalibrated approach, we can use the approximation $\mathbf{T} \simeq \mathbf{I}$, and then

$$V(\bar{\mathbf{x}}) \;\simeq\; \bar{\mathbf{x}}^T\, \mathbf{P}\, \bar{\mathbf{x}}\,. \tag{54}$$

This approximation is exact only if there is no image distortion with respect to the real world (e.g. an affine camera). This is not the case in practice and therefore the computation for the optimal value of $\bar{\alpha}$ will be an approximation. Notice that control stability is independent of this approximation, and this only concerns to the efficiency in terms of performed maneuvers.

Denoting p_{ij} with $i, j = 1, 2, 3$ the entries of matrix $\mathbf{P}$, we can therefore write

$$V(\bar{\mathbf{x}}) \simeq p_{11}\, \bar{\rho}^2 + p_{33}\, \bar{\phi}^2 + p_{22}\, \bar{\alpha}^2 + 2\, p_{23}\, \bar{\alpha}\, \bar{\phi}\,. \tag{55}$$

We search the minimum of this function with respect to $\bar{\alpha}$:

$$\frac{\partial V}{\partial \bar{\alpha}} \simeq 2\, p_{22}\, \bar{\alpha} + 2\, p_{23}\, \bar{\phi} = 0\,. \tag{56}$$

Solving for $\bar{\alpha}$ and using the values of $\mathbf{P}$ we obtain

$$\bar{\alpha} = \bar{\phi}\, \left(\frac{k_\phi^2 - k_\alpha k_\rho + k_\alpha^2}{k_\phi^2 + k_\phi k_\rho + k_\alpha^2} \right)\,. \tag{57}$$

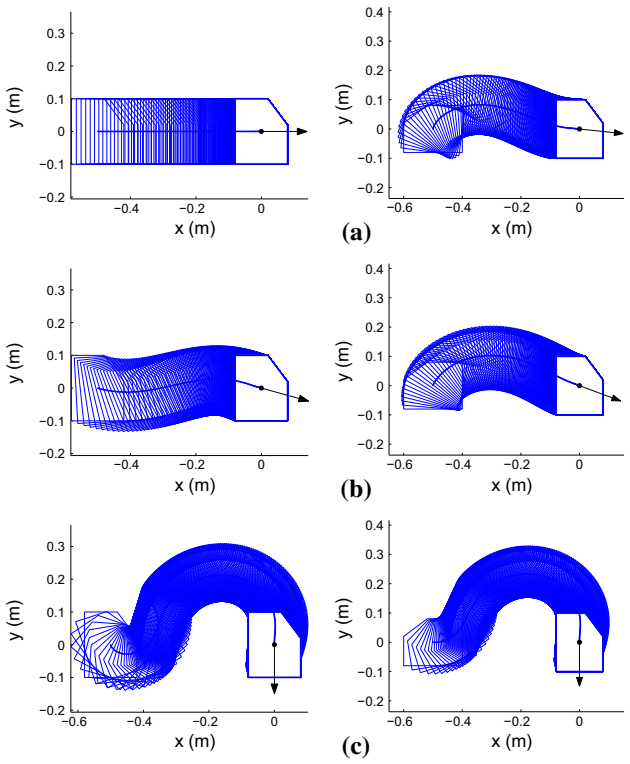

Fig. 7 Two examples, each one in a column, of maneuvers with different pushing directions (denoted with an arrow) on an object. **a** Pushing direction with calibrated data. **b** Pushing direction with uncalibrated data. c Arbitrary pushing direction. The pushing directions are computed in **a** and **b** with the presented procedure. **a** shows an illustrative example where the approximation in (54) is exact, whereas the uncalibrated case is shown in **b**. **c** shows maneuvers by defining an arbitrary pushing direction. It can be seen that examples in **a** and **b** perform better with simple maneuvers

We finally use this solution to assign an approximately optimal motion direction $\mathbf{y}$ to the object. Then, the object will be pushed along this motion direction during the pushing operation. We note that this is computed only once at the beginning of the pushing task and it is set for the rest. Moreover, the stable control law does not violate the forward motion condition (42) while moving the object toward the target during the rest of the task.

Two different examples showing the performance of the object motion when the motion direction is chosen arbitrarily or following the procedure just presented are shown in Fig. 7. The target position in the simulations is $(x, y)^T = (0, 0)^T$ and the initial position is also in both examples $(x, y)^T = (-0.5, 0)^T$. The initial error in ϕ is 0°, and 90° for the left and right column example, respectively. In the first and second rows, the control law is performed with the pushing directions computed as explained above, whereas in the third row they are arbitrarily defined. The results in the first row are computed using undistorted images (so the approximation in (54) is exact). This is provided for illustrative purposes to show the performance with the actual optimal value for the direction to push. The results of the second row are obtained in a general case where an approximation of the optimal values due to (54) are computed. In particular, the camera is located at height of $z = 5$ m and slanted 45°. It can be seen that the first and second rows perform better than the third row in the sense that less maneuvers are required to reach the desired poses. The results in the second row also illustrate that the proposed procedure to select the direction to push is adequate for the uncalibrated framework.

5 Selection of the points to push

Once the motion direction for pushing is computed for a given object, the next step is to choose two pushing points on the contour of the object. These points should allow the object to be pushed along the computed motion direction.

Here we show how to compute these two pushing points on a convex object. Let us define the boundary of the object as a function $\mathbf{B}(\gamma)$ where angle $\gamma \in [0, 2\pi]$ is measured counter-clockwise starting from $\mathbf{y}$. First, we calculate the angle β between the motion direction $\mathbf{y}$ and the inward unit normal vector $\mathbf{n}$ at each point of the object boundary $\mathbf{B}(\gamma)$:

$$\beta(\gamma) = \text{atan2}\,(\,(\mathbf{y} \times \mathbf{n})_z\,,\, \mathbf{y}^T\,\mathbf{n}\,)\,, \quad \gamma = 0 \ldots 2\pi\ . \tag{58}$$

Then, we find the pushing region on the object boundary which satisfies:

$$\beta_{min} < \beta < \beta_{max} \tag{59}$$

where $|\beta_{min}|$ and $|\beta_{max}|$ are equal, and they are defined by the friction coefficient as $\mu = -\tan\beta_{min} = \tan\beta_{max}$. Any point $\mathbf{B}(\gamma)$ that belongs to the pushing region is a possible pushing point candidate. Finally, we choose the left pushing point $\mathbf{P}_L$ and the right pushing point $\mathbf{P}_R$ as follows:

$$\mathbf{P}_L = \mathbf{B}(\gamma_L) \quad \text{with} \quad \gamma_L = \gamma_{min} + \Delta\gamma \tag{60}$$

$$\mathbf{P}_R = \mathbf{B}(\gamma_R) \quad \text{with} \quad \gamma_R = \gamma_{max} - \Delta\gamma \tag{61}$$

where γ_{min} and γ_{max} are the angles of the extremities of the pushing region ($\beta_{min} = \beta(\gamma_{min})$, $\beta_{max} = \beta(\gamma_{max})$); and $\Delta\gamma$ is a small positive angle (e.g., $\Delta\gamma \approx 10°$). Figure 8 shows two different examples of pushing points selection.

6 Experimental results

Manipulation of an object by pushing with two frictional point contacts is validated by experiments. An uncalibrated camera observes the scene from an arbitrary fixed pose. In

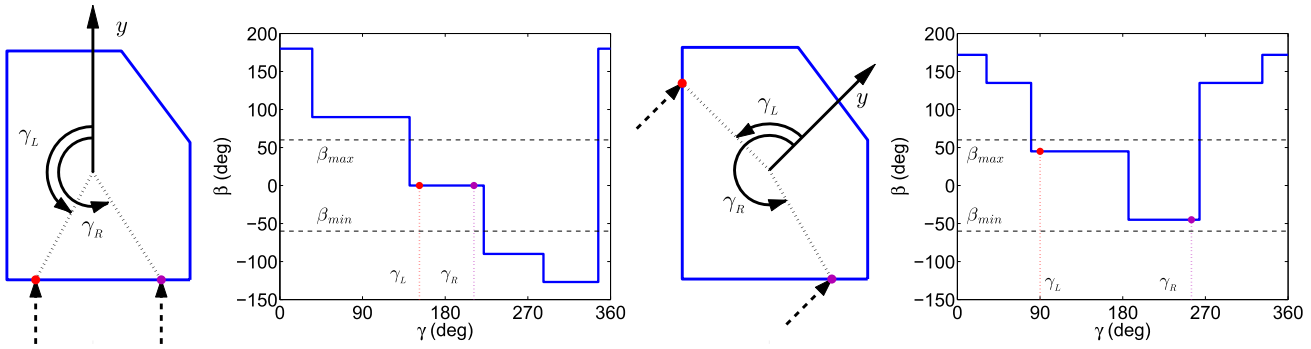

Fig. 8 Examples illustrating the algorithm to select the pushing points in a convex object with two different motion direction **y** (First and third graph, respectively). The pushing points selected are denoted with dashed arrows. Second and fourth graphs show the corresponding alignment angles β of each example. Angle β (58) is defined between the motion direction vector **y** of the object and its inward edge normal vector that varies along the boundary of the object $\mathbf{B}(\gamma)$

the following experiments, we assume that the object has uniform density and therefore the center of mass will be located at the centroid. Experiments are conducted with two different groups of robots. First group is composed of two Khepera non-holonmic mobile robots, and the second group is composed of two Kuka manipulators.

Next sub-sections explain how these different groups of robots are used for pushing, and then present the results of experiments. Additionally, video attachments are also provided to show examples of the control performance of these pushing tasks.

6.1 Pushing with non-holonomic mobile robots

In general, we consider that the actuators can push in any direction. However, in the following set of experiments we use mobile robots as actuators. We remark that a non-holonomic mobile robot can not push the object in every direction. It can push only along its forward motion direction. Therefore, to be able to use non-holonomic mobile robots to manipulate the object, we first choose the two pushing points on the object such that points are at equal distance to the gravity center of the object along the motion direction **y**. Then, the lateral components of the required action (which cannot be produced by the mobile robots) are equal in magnitude and opposite in sign for both robots. Therefore these components are cancelled and the mobile robots are eventually not required to perform lateral actions. This allows us to write the linear speeds of the non-holonomic mobile robots from (5) as below:

$$v_L = v + r_{Lx}\,\omega \tag{62}$$

$$v_R = v + r_{Rx}\,\omega\,. \tag{63}$$

Secondly, the non-holonomic mobile robots may diverge from the pushing points. Therefore we correct the orientation of the non-holonomic mobile robots toward the pushing points (see Fig. 9) with a proportional control given below:

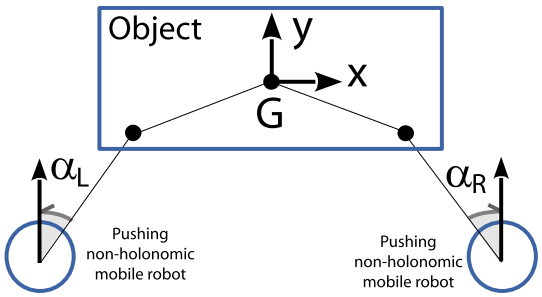

Fig. 9 An object with two non-holonomic mobile robots for pushing

$$\begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = -k_\omega \begin{bmatrix} \alpha_L \\ \alpha_R \end{bmatrix}, \quad k_\omega > 0 \tag{64}$$

where ω_L and ω_R are the computed angular speeds; k_ω is the positive control gain; and α_L and α_R are the orientation errors of the non-holonomic mobile robots with respect to their desired contact points on the object. Since the desired contact point is implicitly used in (64), this control law guarantees that the mobile robot is always pointing toward this contact point. Given that the mobile robots move forward and that they present unicycle motion constraints, this means that the robots will keep contact with the object in the desired points during pushing.

6.2 Pushing with manipulators

Here, we explain how to push an object with a finger-like manipulator tool tip. First, to prevent any divergence from the pushing point, we assign a velocity $\mathbf{v}_T$ to a finger-like

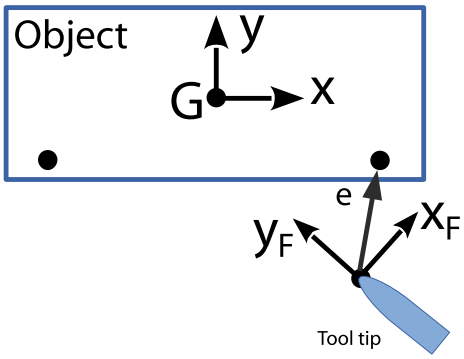

Fig. 10 An object and one of the manipulator tool tips for pushing

tool tip so that it tracks its pushing point:

$$\mathbf{v}_T = -k_e \begin{bmatrix} ((\mathbf{e}^T \mathbf{x})\mathbf{x})^T \mathbf{x}_F \\ ((\mathbf{e}^T \mathbf{x})\mathbf{x})^T \mathbf{y}_F \end{bmatrix}, \quad k_e > 0 \tag{65}$$

where k_e is a positive scalar gain, $\mathbf{e}$ is the vector from the finger-like tool tip to the pushing point, $\mathbf{x}$ is the axis of the object coordinate frame, and $\mathbf{x}_F$ and $\mathbf{y}_F$ are the local coordinate frame axes of the tool tip (see Fig. 10). This tool tip local frame is computed directly from the available image information. Equation (65) keeps a finger-like tool tip in contact with the chosen pushing point. When (65) is integrated with a pushing point velocity computed from (5), the pushing task is performed correctly. This yields:

$$\mathbf{v}_{F_L} = \begin{bmatrix} \mathbf{v}_L^T \mathbf{x}_F \\ \mathbf{v}_L^T \mathbf{y}_F \end{bmatrix} + \mathbf{v}_{T_L} \tag{66}$$

$$\mathbf{v}_{F_R} = \begin{bmatrix} \mathbf{v}_R^T \mathbf{x}_F \\ \mathbf{v}_R^T \mathbf{y}_F \end{bmatrix} + \mathbf{v}_{T_R} \tag{67}$$

where $\mathbf{v}_{F_L}$ and $\mathbf{v}_{F_R}$ are the final velocities of the finger-like left and right tool tips of the manipulators; where $\mathbf{v}_{T_L}$ and $\mathbf{v}_{T_R}$ are the pushing point tracking velocities computed from (65) for the left and right tool tips; and where $\mathbf{v}_L$ and $\mathbf{v}_R$ are the pushing point velocities computed from (5). Later, these final tool tip velocities are transformed to the joint velocities of the manipulators through the known position and velocity kinematics of the manipulators.

6.3 Experiments

6.3.1 Pushing with Khepera non-holonomic mobile robots

First part of the experiments is conducted with two Khepera non-holonomic mobile robots and a box shaped object. The size of the box is 80 × 40 cm. An uncalibrated camera is used to observe the scene from a fixed pose. The camera, connected through Firewire to a computer, is from *Allied Vision Technologies* and mounts a lens with a focal length

Fig. 11 Box shaped object (left) and the two khepera robots (right)

of 3.6 mm. The setup of the robots is located on a table of dimensions 2 × 1.5 m with the camera at a distance about 1.5 m. The size of the robots is 13 cm in diameter and 7 cm in height. We put patterns on the top of the robots and the object so that they can be detected and identified in the images easily (see Fig. 11). Since we proposed uncalibrated image-based visual servoing, we do not compute the robots' poses. The computer that processes the images and computes the control inputs is a laptop with an Intel Core 2 Duo CPU at 2.50 GHz with operating system Ubuntu GNU/Linux (version 8.04). The control scheme runs on this personal computer at some 10 Hz. The size of the acquired images is 1280 × 960 pixels. The code implementation is in C++ and the images are processed with the OpenCV library. The control gains are $k_\rho = 0.2$, $k_\alpha = 1$, $k_\phi = 0.4$, and $k_\omega = 2$. The control velocities are sent to the mobile robots through wireless Ethernet network connection.

Figures 12, 13 and 14 show the results of three different experiments where the proposed control law pushes the object from three different poses to the same target pose. In each figure, the desired and initial configurations of the box are shown in the images of the top row. The second row shows the object in the final configuration after running the control algorithm with the plot of the resultant robots' path. Third row presents the input velocities of the robots (v_L, ω_L), and (v_R, ω_R), computed from (62–63) and (64). Notice that the maximum allowed velocity of the robots was set to 10 cm/s, so the output control values are saturated to this value. The last row shows the evolution of the state variables of the object $\bar{\rho}$ and $\bar{\phi}$ during the pushing operation. From these results, one can see that the object is placed to the target pose correctly. Although we assume non-slip contact between the robots and the object, varying distances can be seen between the two pushers along the trajectories. This is a practical issue due to system perturbations, such as the initial acceleration of the pusher robots. Nevertheless, it can be seen that the system recovers properly during the motion even if the transient is long due to the low actions and slow motions. Note that motions are required to be slow enough so that inertial forces are negligible.

6.3.2 Pushing with Kuka dual-arm manipulator

Second part of the experiments is conducted with Kuka dual-arm manipulator platform and a box shaped object. The

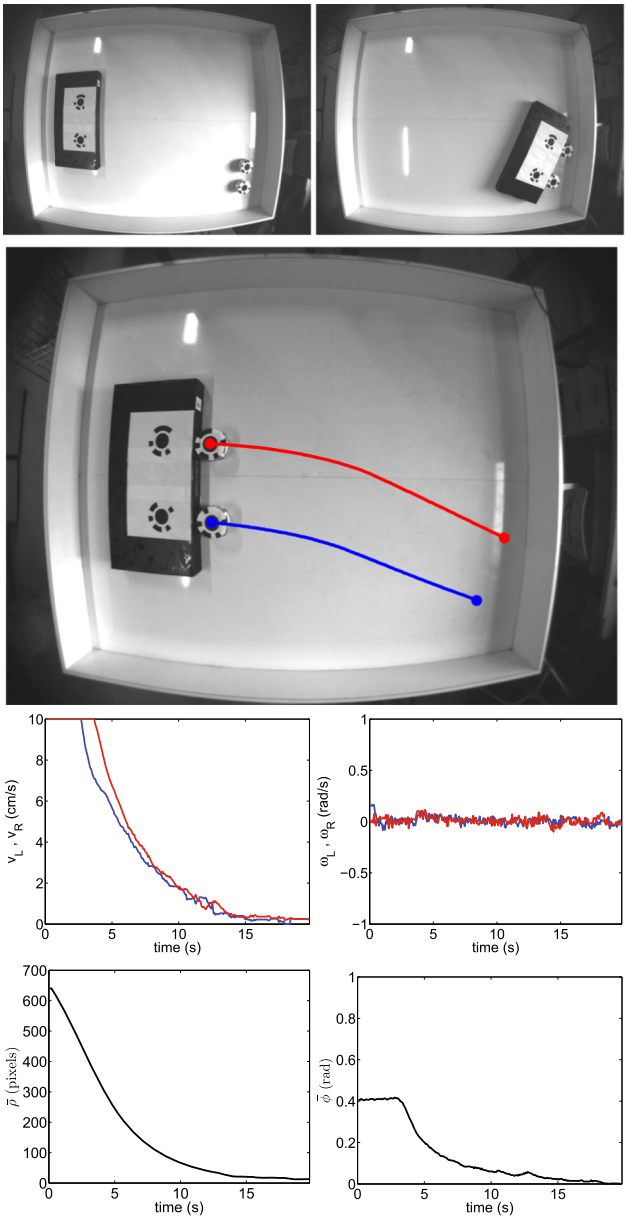

Fig. 12 1st row: Target (left) and initial (right) object poses with two pushing non-holonomic mobile robots. 2nd row: Performed pushing manipulation and robots' trajectories. 3rd row: Evolution of the robots' velocities. 4th row: State variables $\bar{\rho}$ and $\bar{\phi}$ of the object versus time

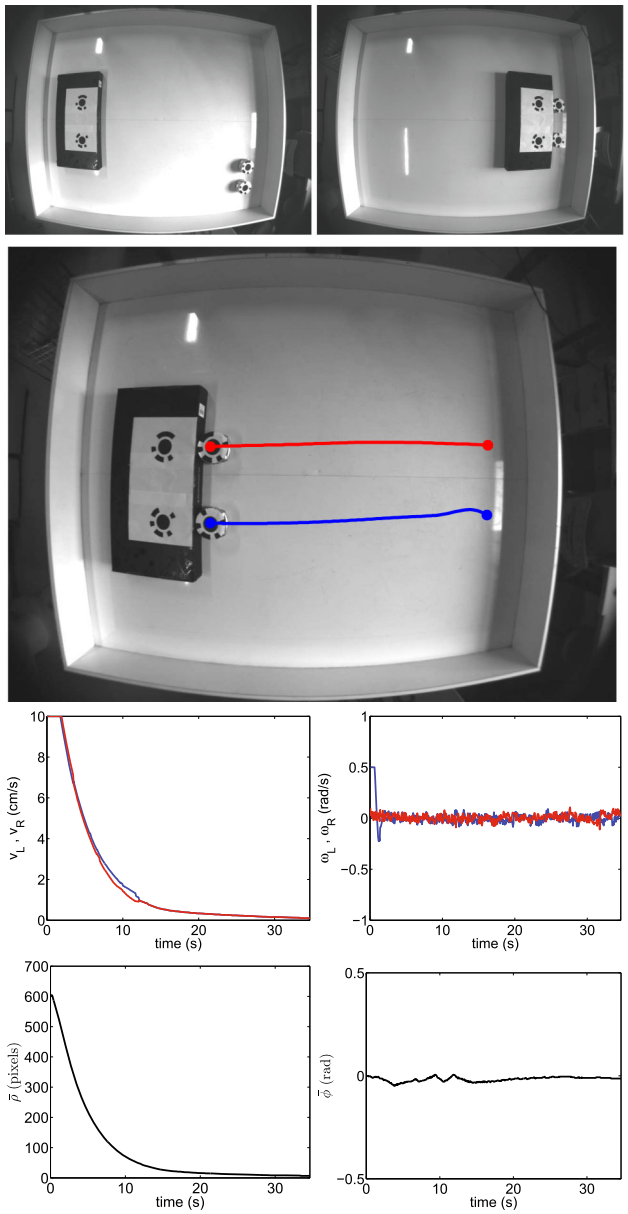

Fig. 13 1st row: Target (left) and initial (right) object poses with two pushing non-holonomic mobile robots. 2nd row: Performed pushing manipulation and robots' trajectories. 3rd row: Evolution of the robots' velocities. 4th row: State variables $\bar{\rho}$ and $\bar{\phi}$ of the object versus time

setup of the robots is located on a table of dimensions about 80 × 200 cm. The size of the box is 60 × 40 cm. Kuka dual-arm manipulator platform has a stereo camera head. The camera model is *Imaging Source DFK 31AF03-Z2* and belongs to the *TO40* Stereo Pan-Tilt Head from *Robosoft*. One of these uncalibrated Firewire cameras is used to observe the scene from a distance around 1.5 m (see Fig. 1). This camera has zooming degree of freedom so its focal changes between 5 and 45 mm. The Kuka dual-arm manipulator platform is connected to a personal computer. The control scheme runs on this personal computer at some 25 Hz. The size of the acquired images is 1024 × 768 pixels. The computer model used is a *Dell Precision T3600* with processor *Intel Xeon* at 3.60 GHz, and Ubuntu GNU/Linux operating system (version 12.04). The code implementation is again in C++ although now the images are processed with the ViSP library (Marchand et al. 2005). The control gains are $k_\rho = 0.03, k_\alpha = 0.15, k_\phi = 0.05$, and $k_e = 0.003$. The computed control velocities are sent to the pushing hand actuators of the Kuka dual-arm manipulator using the robot model.

Figures 15 and 16 show the results for a performed pushing manipulation of the box shaped object with the Kuka dual-

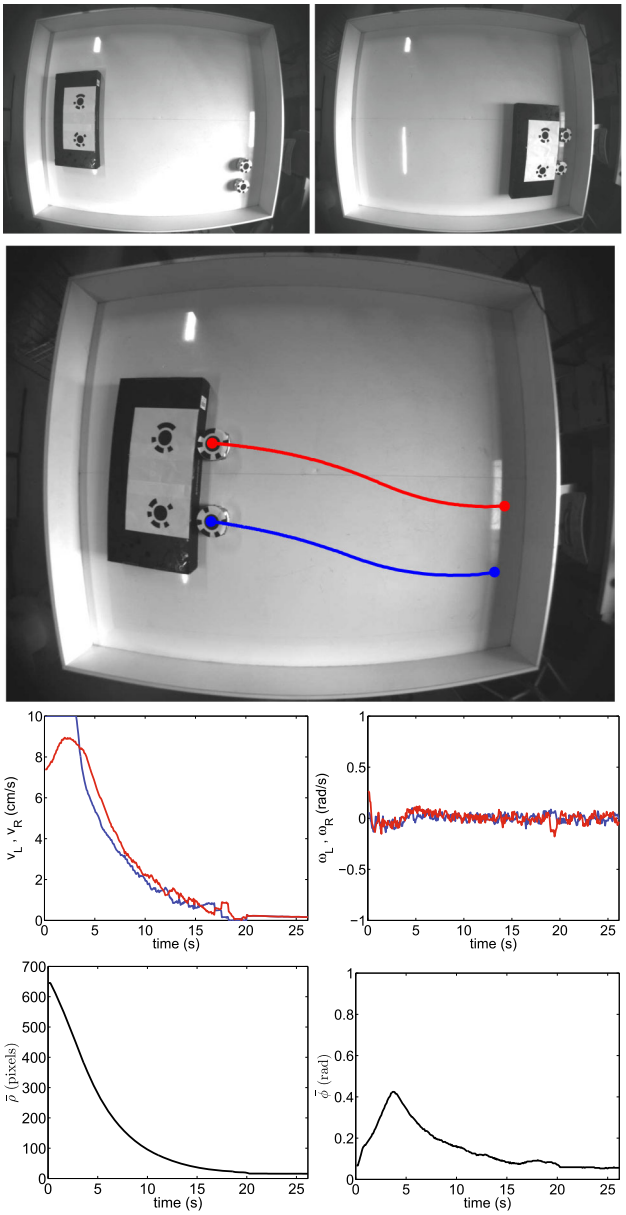

Fig. 14 1st row: Target (left) and initial (right) object poses with two pushing non-holonomic mobile robots. 2nd row: Performed pushing manipulation and robots' trajectories. 3rd row: Evolution of the robots' velocities. 4th row: State variables $\bar{\rho}$ and $\bar{\phi}$ of the object versus time

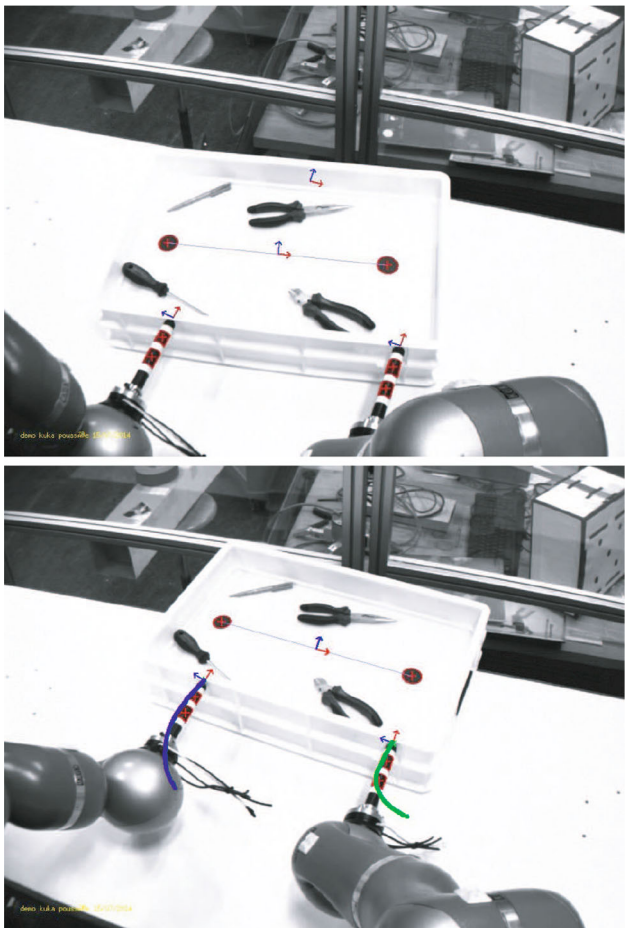

Fig. 15 Top image shows the initial pose of the dual-arm manipulator, the box shaped object before manipulation, and the target pose frame of the object. Bottom image shows the performed pushing manipulation and the traces (blue and green) of the tool tips of the dual-arm manipulator during pushing (Color figure online)

arm manipulator platform. The images in Fig. 15 shows the initial pose of the object and the the final pose when the target configuration is reached. In this manipulation, the initial values of the state variables of the object were as follows: $\bar{\rho} = 127$ pixels, $\bar{\alpha} = -10°$ and $\bar{\phi} = 10°$. First row in Fig. 16 shows the evolution of the velocities of the tool tips of the manipulators computed from (66) and (67) with the proposed control law. Second row shows the evolution of the state variables $\bar{\rho}$ and $\bar{\phi}$ of the object versus time. The results show that the control performs properly and that the object converged to the target pose correctly.

7 Conclusion

This paper proposed an uncalibrated image-based visual servoing scheme to position objects on the plane by pushing with two mobile robots or dual-arm manipulator. Standard approaches usually require performing a number of discrete maneuvers to guarantee reaching the target, whereas our approach is smooth in the sense that the motion is performed in one go without chaining several maneuvers. As can be seen in the results, trajectories performed by the pushed objects are smooth and continuous. Moreover, performing several maneuvers would be specially inefficient in our framework since it would require to repeat the control execution several times, each time defining the new pushing points (and translating the pushing actuators to them). In our approach, the key point to avoid unnecessary maneuvers is that we can initially choose the direction to push that minimizes the initial energy of the system in the sense of Lyapunov. This

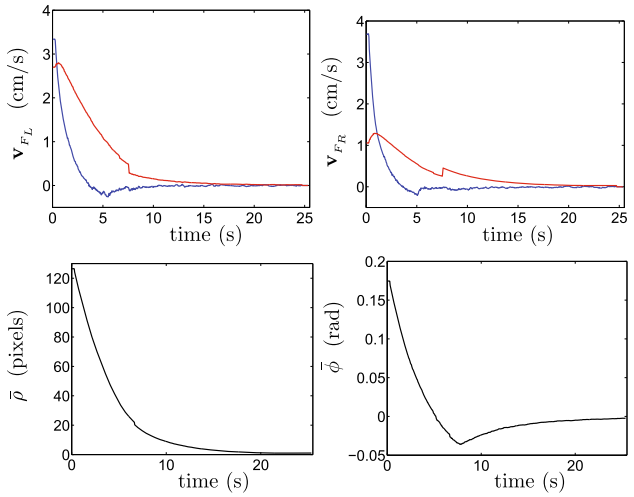

Fig. 16 Results of the manipulation performed in Fig. 15. First row shows the evolution of the velocities of the tool tips of the manipulators computed from (66) and (67). The velocity of each tool tip consists of a two dimensional vector where the x-component is plotted in red and the y-component in blue. Second row shows the state variables $\bar{\rho}$ and $\bar{\phi}$ of the object versus time (Color figure online)

improves the efficiency because the cost over the entire trajectory is reduced producing a more direct motion to the target.

The stability of the control law of this uncalibrated approach is also proven. Regarding the workspace of the approach, there are no constraints for the possible desired poses of the object as far as it can be appropriately pushed, i.e. the direction to push computed by the method can be applied in practice. These limits are defined by the considered assumptions. In particular, we constrain the setup for convex objects, and we impose that no slip occurs during the pushing. One of the conditions for the feasibility of the proposed approach is the quasi-static assumption, which is reasonable as long as the dynamic forces are negligible. Here, this assumption can be easily ensured by using appropriate control gains to provide low control signals, but also with enough thrust to overcome dead zones. The correctness of the proposed control scheme is validated by several experiments. Furthermore, this approach is easy to put in practice since it is uncalibrated, and it is robust to modeling errors since it is closed directly in the sensor space.

Acknowledgements This work was supported by French Government research program Investissements d'avenir through the RobotEx Equipment of Excellence (ANR-10-EQPX-44), the LabEx IMobS3 (ANR7107LABX716701) and I-SITE Project (CAP 20-25) and by Spanish Government/European Union through Project DPI2015-69376-R (MINECO/FEDER).

References

Adachi, J., & Sato, J. (2004). Uncalibrated visual servoing from projective reconstruction of control values. In *17th international conference on pattern recognition, ICPR* (pp. 297–300).

Akella, S., & Mason, M. T. (1998). Posing polygonal objects in the plane by pushing. *The International Journal of Robotics Research*, *17*(1), 70–88.

Akella, S., & Mason, M. T. (1999). Using partial sensor information to orient parts. *International Journal of Robotics Research*, *18*(10), 963–997.

Balorda, Z., & Bajd, T. (1994). Reducing positioning uncertainty of objects by robot pushing. *IEEE Transactions on Robotics and Automation*, *10*(4), 535–541.

Amor, H. B., Saxena, A., Hudson, N., & Peters, J. (2014). Special issue on autonomous grasping and manipulation. *Autonomous Robots*, *36*, 1–3.

Brost, R. C. (1986). Automatic grasp planning in the presence of uncertainty. *IEEE International Conference on Robotics and Automation*, *3*, 1575–1581.

Cai, C., Somani, N., & Knoll, A. (2016). Orthogonal image features for visual servoing of a 6-dof manipulator with uncalibrated stereo cameras. *IEEE Transactions on Robotics*, *32*(2), 452–461.

Caron, G., Marchand, E., & Mouaddib, E. M. (2013). Photometric visual servoing for omnidirectional cameras. *Autonomous Robots*, *35*, 177–193.

Chaumette, F., & Hutchinson, S. (2006). Visual servo control, part I: Basic approaches. *IEEE Robotics and Automation Magazine*, *13*(4), 82–90.

Chen, H., Sun, D., & Yang, J. (2009). Global localization of multirobot formations using ceiling vision SLAM strategy. *Mechatronics*, *19*(5), 617–628.

Chesi, G., & Hashimoto, K. (Eds.) (2010). *Visual servoing via advanced numerical methods* (Vol. 401). Lecture notes in control and information sciences. Heidelberg: Springer.

Dogar, M., & Srinivasa, S. (2011). A framework for push-grasping in clutter. In *Proceedings of robotics: Science and systems*.

Gandolfo, F., Tistarelli, M., & Sandini, G. (1991). Visual monitoring of robot actions. In *Workshop on intelligence for mechanical systems. IEEE/RSJ international conference on intelligent robots and systems* (Vol. 1, pp. 269–275).

Golkar, M.A., Namin, S.T., & Aminaiee, H. (2009). Fuzzy controller for cooperative object pushing with variable line contact. In *IEEE international conference on mechatronics* (pp. 1–6).

Jara, C.A., Pomares, J., F.A. Candelas Herías, & Torres, F. (2014). Optimal control for robot-hand manipulation of an object using dynamic visual servoing. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 89–94).

Kopicki, M., Zurek, S., Stolkin, R., Moerwald, T., & Wyatt, J. L. (2016). Learning modular and transferable forward models of the motions of push manipulated objects. *Autonomous Robots*, *41*, 1–22.

Li, B., Fang, Y., & Zhang, X. (2014). Projection homography based uncalibrated visual servoing of wheeled mobile robots. In *IEEE conference on decision and control* (pp. 2167–2172).

Li, Q., & Payandeh, S. (2007). Manipulation of convex objects via two-agent point-contact push. *The International Journal of Robotics Research*, *26*(4), 377–403.

Liang, X., Wang, H., Chen, W., Guo, D., & Liu, T. (2015). Adaptive image-based trajectory tracking control of wheeled mobile robots with an uncalibrated fixed camera. *IEEE Transactions on Control Systems and Technology*, *23*(6), 2266–2282.

López-Nicolás, G., Özgür, E., & Mezouar, Y. (2015). Image-based control of two mobile robots for object pushing. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 5472–5478).

López-Nicolás, G., & Sagüés, C. (2011). Vision-based exponential stabilization of mobile robots. *Autonomous Robots*, *30*, 293–306.
Lynch, K. M. (1992). The mechanics of fine manipulation by pushing. *IEEE International Conference on Robotics and Automation*, *3*, 2269–2276.
Lynch, K. M., Maekawa, H., & Tanie, K. (1992). Manipulation and active sensing by pushing using tactile feedback. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, *1*, 416–421.
Lynch, K. M., & Mason, M. T. (1996). Stable pushing: Mechanics, controllability, and planning. *The International Journal of Robotics Research*, *15*(6), 533–556.
MacMillan, W. D. (1936). *Dynamics of rigid bodies*. NY: Dover.
Malis, E. (2004). Visual servoing invariant to changes in camera-intrinsic parameters. *IEEE Transactions on Robotics and Automation*, *20*(1), 72–81.
Marchand, E., Spindler, F., & Chaumette, F. (2005). Visp for visual servoing: A generic software platform with a wide class of robot control skills. *IEEE Robotics Automation Magazine*, *12*(4), 40–52.
Mason, M. T. (1986). Mechanics and planning of manipulator pushing operations. *The International Journal of Robotics Research*, *5*(3), 53–71.
Mason, M. T. (2001). *Mechanics of robotic manipulation*. Cambridge: MIT Press.
Meriçli, T., Veloso, M., & Akın, H. V. (2015). Push-manipulation of complex passive mobile objects using experimentally acquired motion models. *Autonomous Robots*, *38*, 317–329.
Nammoto, T., Hashimoto, K., Kagami, S., & Kosuge, K. (2013). High speed/accuracy visual servoing based on virtual visual servoing with stereo cameras. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 44–49).
Okawa, Y., & Yokoyama, K. (1992). Control of a mobile robot for the push-a-box operation. *IEEE International Conference on Robotics and Automation*, *1*, 761–766.
Park, J. S., & Chung, M. J. (2003). Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy. *IEEE Transactions on Robotics and Automation*, *19*(2), 250–258.
Peshkin, M. A., & Sanderson, A. C. (1988). The motion of a pushed, sliding workpiece. *IEEE Journal of Robotics and Automation*, *4*(6), 569–598.
Peshkin, M. A., & Sanderson, A. C. (1988). Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal of Robotics and Automation*, *4*(5), 524–531.
Piepmeier, J. A., McMurray, G. V., & Lipkin, H. (2004). Uncalibrated dynamic visual servoing. *IEEE Transactions on Robotics and Automation*, *20*(1), 143–147.
Ramirez, O.A., & Jägersand, M. (2016). Practical considerations of uncalibrated visual servoing. In *13th conference on computer and robot vision, CRV* (pp. 164–169).
Rezzoug, N., & Gorce, P. (1999). Dynamic control of pushing operations. *Robotica*, *17*(6), 613–620.
Rus, D., Donald, B., & Jennings, J. (1995). Moving furniture with teams of autonomous robots. In *IEEE/RSJ international conference on intelligent robots and systems* (Vol. 1, pp. 235–242).
Salganicoff, M., Metta, G., Oddera, A., & Sandini, G. (1993). A direct approach to vision guided manipulation. In *International conference on advanced robotics*.
Shademan, A., & Jägersand, M. (2012). Robust sampling-based planning for uncalibrated visual servoing. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 2663–2669).
Sudsang, A., Rothganger, F., & Ponce, J. (2002). Motion planning for disc-shaped robots pushing a polygonal object in the plane. *IEEE Transactions on Robotics and Automation*, *18*(4), 550–562.

Gonzalo López-Nicolás received the degree in Industrial Engineering and the degree of Ph.D. in Systems Engineering and Computer Science from the University of Zaragoza, Spain. He is Associate Professor with the Department of Computer Science and Systems Engineering at the Universidad de Zaragoza. He is member of the Robotics, Perception and Real-Time Group (RoPerT), and the Instituto de Investigación en Ingeniería de Aragón (I3A). His current research interests are focused on visual control, autonomous robot navigation, multirobot systems, and the application of computer vision techniques to robotics.

Erol Özgür received Ph.D. degree in Robotics and Vision from Université Blaise Pascal in 2012. He was a postdoctoral fellow in Institut Pascal - UBP/CNRS/IFMA between 2012 and 2014. Since 2015, he conducts research and teaching in Université Clermont Auvergne. His research interests are vision-based robot control and computer vision.

Youcef Mezouar received the Ph.D. degrees in automation and computer science from the University of Rennes 1, France in 2001. He spent one year as Postdoctoral Associate in the Robotic Lab of the Computer Science Department of Columbia University, New York. He was assistant professor from 2002 to 2011 in the Physics Department of Blaise Pascal University, Clermont-Ferrand, France. He holds a tenured position as Full Professor at SIGMA Clermont (IFMA until 2016) since 2012. At SIGMA, he is the head of the Machines, Mechanisms and Systems Department and he currently leads the MACCS (Modeling, Autonomy and Control of ComplEx System) team at Institut Pascal. His research interests include vision-based and multisensor-based control.