

# Vision-based navigation of omnidirectional mobile robots

Marco Ferro, Antonio Paolillo, Andrea Cherubini and Marilena Vendittelli

**Abstract**—This paper considers the problem of collision-free navigation of omnidirectional mobile robots in environments with obstacles. Information from a monocular camera, encoders, and an inertial measurement unit is used to achieve the task. Three different visual servoing control schemes, compatible with the class of considered robot kinematics and sensor equipment, are analysed and their robustness properties with respect to actuation inaccuracies discussed. Then, a controller is proposed with formal guarantee of convergence to the bisector of a corridor. The main controller components are a visual servoing control scheme and a velocity estimation algorithm integrating visual, kinematic and inertial information. The behaviour of all the considered algorithms is analysed and illustrated through simulations both for a wheeled and a humanoid robot. The solution proposed as the most efficient and robust with respect to actuation inaccuracies is also validated experimentally on a real humanoid NAO.

**Index Terms**—Visual-Based Navigation, Visual Servoing, Collision Avoidance

## I. INTRODUCTION

MOBILE robots have, by construction, an unlimited workspace wherein to move safely to accomplish tasks in industrial and service contexts. The working environments are typically wide, dynamic and difficult, or impossible, to structure.

Examples include robotic tasks in logistics or office-like environments, typically cluttered and dynamic, or exploration and rescue tasks in post-disaster environments, highly unstructured with difficult deployment of external sensory services. In these situations, a reactive navigation mode relying solely on on-board sensors and exploiting the whole-body reconfiguration capabilities of the robot is desirable.

Omnidirectional robots, both wheeled and legged, offer a high degree of manoeuvrability that can ease navigation in challenging scenarios, like moving sideways in narrow

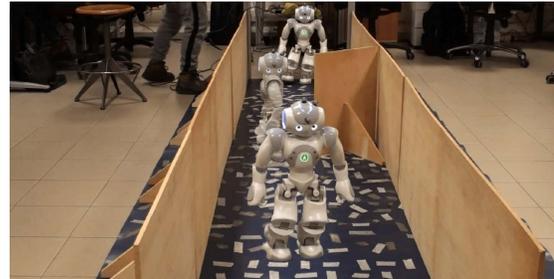


Fig. 1: NAO going through a narrow passage by walking sideways.

passages (see Fig. 1). In conjunction with vision-based control methods, these robots can achieve the above mentioned tasks in an effective and safe way.

Within vision-based control methods visual servoing (IBVS) [1] techniques are particularly suited to working conditions requiring a reactive robot behaviour because they define the task in the sensor space, without the need of a map nor of the robot pose estimation within the environment. In addition, these methods are quite effective in case of monocular cameras, almost always available also in the sensor suite of cheap mobile robots.

In this paper we adopt the visual servoing paradigm to keep omnidirectional mobile robots clear of the surrounding obstacles through the regulation of properly defined features in the image plane of the robot camera.

### A. Related work

Despite the success of humanoids in recent years and the wide use of omnidirectional wheeled robots in the industrial context, few works address the problem of omnidirectional navigation among obstacles. An obstacle avoidance navigation scheme, where a number of drivable paths are defined to generate the omnidirectional motion and avoid obstacles, while reaching a visible target, is proposed in [2]. The method uses laser scans to perceive the surrounding environment and does not exploit visual information. The navigation scheme proposed in [3] solves a simultaneous localization and mapping problem, while generating an optimised trajectory that exploits the efficient manoeuvrability of the omnidirectional platform. The motion is planned in advance, therefore no reactive behaviour is possible with respect to unpredicted obstacles. A reactive obstacle avoidance behaviour is obtained in [4] based on optical flow clustering and an optimization process. Although capable of avoiding static and moving obstacles in wide workspaces, the method does not exploits the high maneuverability of the vehicle to go through narrow passages in cluttered scenarios.

Manuscript received: December, 6, 2018; Revised March, 11, 2019; Accepted April, 8, 2019.

This paper was recommended for publication by Editor Eric Marchand upon evaluation of the Associate Editor and Reviewers' comments. \*This work is supported by the European Commission, within the H2020 COMANOID project (<http://comanoid.cnrs.fr>).

Marco Ferro is with Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Roma, Italy, [ferro@diag.uniroma1.it](mailto:ferro@diag.uniroma1.it)

Antonio Paolillo was previously with Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM), Université de Montpellier, CNRS, 34095 Montpellier Cedex 5, France. He is currently with Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland, [antonio.paolillo@idiap.ch](mailto:antonio.paolillo@idiap.ch)

Andrea Cherubini is with LIRMM, Université de Montpellier, CNRS, 34095 Montpellier Cedex 5, France, [cherubini@lirmm.fr](mailto:cherubini@lirmm.fr)

Marilena Vendittelli is with Dipartimento di Ingegneria dell'Informazione, Elettronica e Telecomunicazioni, Sapienza Università di Roma, via Eudossiana 18, 00184 Roma, Italy, [marilena.vendittelli@uniroma1.it](mailto:marilena.vendittelli@uniroma1.it)

Digital Object Identifier (DOI): see top of this page.

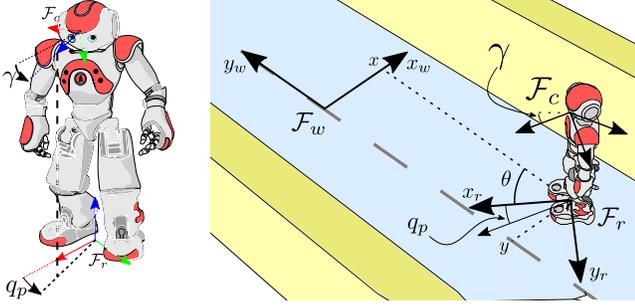


Fig. 2: Frames of interest to problem formulation.

On the other hand, vision-based navigation of humanoid robots is widely studied. A comprehensive description of the general problem is given in [5]. The authors of [6] present vision-based navigation with obstacles detection for a humanoid robot in a learning fashion. The robot detects and classifies the obstacles in order to identify the traversable space. However, the detection also requires the use of laser data, along with monocular camera images. In [7], a vision-based navigation scheme has been presented for a humanoid robot in a maze-like environment. The robot processes the images from the camera to extract a pair of visual guidelines that identify the navigating corridor, and then uses an IBVS control scheme to walk at the center of the corridor. The scheme considers the robot moving according to a unicycle motion model, does not exploit the omnidirectional walking of humanoids to go through narrow passages, and assumes that the corridor is large enough to allow safe navigation and is obstacle-free. In [8], robot navigation is addressed through a planner based on a pre-built visual memory. The planner takes into account the presence of unexpected obstacles and exploits the omnidirectional motion capabilities of the humanoid to avoid obstacles. However, since the robot is not able to look towards its moving direction, there is no guarantee that other unseen obstacles are avoided.

### B. Contribution of the paper

With respect to the approaches described above, this paper proposes a reactive vision-based navigation scheme for omnidirectional robots, that does not rely on map building nor on localization methods, and takes into account possible uncertainties on the robot velocity. Exploiting omnidirectional mobility and kinematic redundancy, the mobile robots are kept clear of obstacles by moving in the direction of their camera gaze, while orienting the body in accordance to the size of the traversable space.

With respect to [9], in this paper we show that the gaze-velocity alignment, enforced in [9], represents a steady state constraint in pure visual servoing control schemes that suit the considered robot kinematics and sensors equipment.

Through a formal comparison of these IBVS schemes we show that a single, appropriate feature and gaze-velocity alignment, as proposed in [9], not only is sufficient to ensure convergence of the robot to the bisector of a corridor but also ensures better transient behaviour.

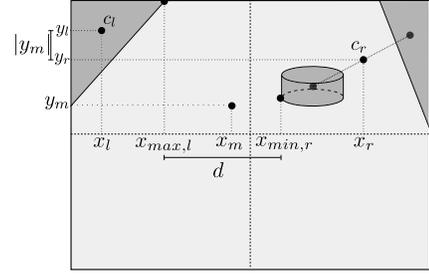


Fig. 3: Graphical illustration of the visual features  $x_m$ ,  $y_m$  defined by (1), and of the distance  $d$  used in eq. (5).

The accurate satisfaction of the gaze-velocity alignment constraint, however, is critical for the controller convergence. Therefore, an open loop velocity command is not reliable. To solve the problem in this paper we estimate the robot actual velocity to be fed back in the control loop.

The paper is organised as follows: Section II formulates the problem and summarises the approach proposed in [9]. Section III presents the considered vision-based strategies and discusses their ability to cope with velocity uncertainties. Section IV describes the design of the Kalman filter for velocity estimation and the resulting controller making use of velocity feedback. Section V presents simulations and experiments and Section VI concludes the paper.

## II. PROBLEM FORMULATION AND PROPOSED APPROACH

Consider an omnidirectional (wheeled or legged) mobile robot that has to safely navigate within an unknown and cluttered environment, with flat ground. The robot can rely on a monocular camera, an IMU and encoders at joints. Figure 2 shows the frames of interest:  $\mathcal{F}_w$ , the world inertial frame;  $\mathcal{F}_c$ , the camera frame with  $z$ -axis along the camera focal axis and the  $y$ -axis pointing downwards;  $\mathcal{F}_r$ , the robot frame with the  $x$ -axis directed along the forward direction of motion. The orientation of  $\mathcal{F}_c$  w.r.t.  $\mathcal{F}_r$  is expressed through a tilt and a pan angle. The first, denoted by  $\gamma$ , is kept fixed, while the latter, denoted by  $q_p$ , is an additional degree of freedom exploited by the navigation algorithm to set the direction of motion. The camera position, projected on the ground, in  $\mathcal{F}_w$  is denoted by  $x$  and  $y$ , whereas  $\theta$  is the orientation of the robot body with respect to  $y_w$ . The camera orientation in  $\mathcal{F}_w$  is  $\theta_c = \theta + q_p$ .

The robot commands available for the navigation task are given by the velocity vector  $\mathbf{v}_r = (v_x, v_y, \omega)^T$ , composed by the translational velocity components  $v_x$  and  $v_y$ , and the rotational velocity  $\omega$ , around  $z_r$ , all expressed in  $\mathcal{F}_r$ . Furthermore, the camera pan angle is actuated through the joint velocity command  $u_p$ , such that  $u_p = \dot{q}_p$ .

To achieve the control objective of safe navigation, we consider the regulation on the image plane of proper visual features, defined as functions of all the obstacle centroids detected on the image. We consider as obstacle each connected region in the image, not belonging to the *dominant plane*, which is determined by the image processing procedure in [10]. This procedure yields a mask distinguishing ground (light gray in Fig. 3) from obstacle (dark gray in Fig. 3) pixels. Denoting by  $\mathbf{c}_l = (x_l, y_l)$  and  $\mathbf{c}_r = (x_r, y_r)$  the average of the obstacle

centroids in the left (respectively, right) half of the image, we define the visual features (Fig. 3):

$$x_m = \frac{1}{2}(x_l + x_r) \quad , \quad y_m = (y_l - y_r) \quad (1)$$

where  $x_m$  denotes the abscissa of the middle point between  $c_l$  and  $c_r$ , while  $y_m$  is the difference of ordinates. Note that (1) is ill-conditioned if there are no obstacles in one or both halves of the image. Therefore, we place one/two fictitious centroid/s on the image border of the centroid-free half/ves, to keep  $x_m$  and  $y_m$  well-defined. While this alters the true visual features dynamics, it provides a conservative behaviour when obstacles are out the camera field of view, preserving the effectiveness of the navigation both in structured scenes (e.g., 90-degree-turns, T-junctions, see [9]), and in more general unstructured cluttered scenarios.

The dynamics of the visual features (1) is a function of  $v_r$  and  $u_p$ :

$$\begin{aligned} \begin{pmatrix} \dot{x}_m \\ \dot{y}_m \end{pmatrix} &= \begin{pmatrix} J_{v_x}^x & J_{v_y}^x & J_{\omega_z}^x \\ J_{v_x}^y & J_{v_y}^y & J_{\omega_z}^y \end{pmatrix} v_r + \begin{pmatrix} J_{\omega_z}^x \\ J_{\omega_z}^y \end{pmatrix} u_p = \\ &= \begin{pmatrix} J_x \\ J_y \end{pmatrix} v_r + J_{\omega_z} u_p = J v_r + J_{\omega_z} u_p, \end{aligned} \quad (2)$$

where  $J$  and  $J_{\omega_z}$ , are the image Jacobians with respect to the considered velocity inputs [9]. The idea is to regulate to zero the visual features while walking. If the motion is aligned with the camera gaze, this corresponds to maximise the robot distance to the obstacles in the workspace. In particular, in case of a corridor, we have the following:

**Proposition 1** [9]: *Assuming navigation along a corridor without obstacles, the regulation of the visual feature  $x_m$  to zero, implies the convergence of the camera position and orientation to the corridor bisector, i.e.:*

$$x_m \rightarrow 0 \implies (x, \theta_c)^T \rightarrow (0, 0)^T, \quad (3)$$

if and only if the following constraint is satisfied:

$$\begin{cases} v_x = v \cos q_p \\ v_y = v \sin q_p \end{cases} \quad (4)$$

with  $v = \sqrt{v_x^2 + v_y^2}$  the norm of the robot linear velocity. Note that, the proof of this proposition in [9], also provides a lower bound on the value of  $\gamma$  (i.e., the camera has to be sufficiently tilted towards the floor).

Proposition 1 guarantees convergence of the camera to the corridor bisector, thus maximising distance of the camera from obstacles. However, depending on the robot footprint, this does not guarantee obstacle avoidance by the robot body, particularly in narrow passages. To allow negotiation of narrow passages, the algorithm uses the image distance  $d = |x_{max,l} - x_{min,r}|$  between right and left obstacle contours, being  $x_{max,l}$  ( $x_{min,r}$ ) the abscissa of the rightmost (leftmost) pixel of left (right) obstacles (see Fig. 3). This value is related to the measure of the passage width, used to derive a reference value  $q_p^*$  for the camera pan angle:

$$q_p^* = \begin{cases} \pi/2 & \text{if } d < d_{\min} \\ \frac{\pi}{2} \left(1 - \frac{d - d_{\min}}{d_{\max} - d_{\min}}\right) & \text{if } d_{\min} \leq d \leq d_{\max} \\ 0 & \text{if } d > d_{\max}, \end{cases} \quad (5)$$

with  $d_{\min}$  and  $d_{\max}$  given hysteresis lower and upper values, that prevent chattering effects due to image processing noise. According to (5), if  $d$  is sufficiently high (i.e., the passage is considered large enough)  $q_p^* = 0$ , thus the robot orientation is kept aligned with the camera. On the other hand, if  $d < d_{\max}$  (the passage becomes narrow)  $q_p^*$  linearly changes from 0 to  $\pi/2$  and the robot starts walking sideways by enforcing constraint (4). It is worth noticing that the values of  $q_p^*$  have been set assuming the robot motion along the shorter dimension of its platform footprint, as typically happens for humanoids or human-like wheeled robots [11]. However, determining the  $q_p^*$  that best suits other robot reconfiguration needs is out of the paper scope and, in our envisioned architecture, is the prerogative a higher decisional level.

A nice consequence of the above result is that, since the corridor walls are not required to be parallel, in a maze of corridors the robot will navigate on the Voronoi diagram of the environment, provided that the transient phases occurring at the transitions between corridors are short. The direct control of the robot velocity ensures short transient in the gaze-velocity alignment dynamics, thus minimising risk of collisions in these phases.

Note, however, that the corridor scenario is used only as a benchmark (for comparison with, e.g., [7]) and to prove formal properties. Since the perception part does not assume any structure in the environment, the strategy is effective also for navigation in generic environments with obstacles as will be shown through simulations and experiments.

Summarising, the desired navigation task is achieved by accomplishing the following tasks: (i) align the robot velocity direction with the camera gaze (i.e., satisfy (4)); (ii) regulate the visual features to zero; (iii) regulate the camera pan angle to the desired values to allow sidewalk in narrow passages.

### III. VISION-BASED CONTROL STRATEGIES

This section presents three controllers that regulate to zero the error  $e = (e_v, e_p)^T$ , where  $e_v$  is a visual error defined to achieve task (ii), and  $e_p = q_p - q_p^*$  is used to achieve task (iii). The three controllers differ in the definition of  $e_v$  and in the way to achieve task (i), i.e., to enforce the constraint (4).

#### A. 1D-IBVS

The approach, proposed in [9], defines the visual error as  $e_v = x_m$ . Exponential convergence of  $e_v$  to zero is guaranteed by the control law:

$$\begin{aligned} \omega &= -J_{\omega}^{-1} \left( K_{x_m} x_m + J_{v_x}^x v_x + J_{v_y}^x v_y \right) - u_p \\ u_p &= -K_p e_p. \end{aligned} \quad (6)$$

In the above equation,  $K_{x_m}$  and  $K_p$  are two positive gains, and  $v_x$  and  $v_y$  are given by (4). The controller (6) is able to successfully achieve the navigation task assuming that the constraint enforced by the commanded velocity (4) is actually satisfied. However, in real conditions, the actual velocity of the robot may differ from the nominal one. Therefore, (6) may fail to achieve the control objectives if the real values of  $v_x$  and  $v_y$  do not satisfy constraint (4).

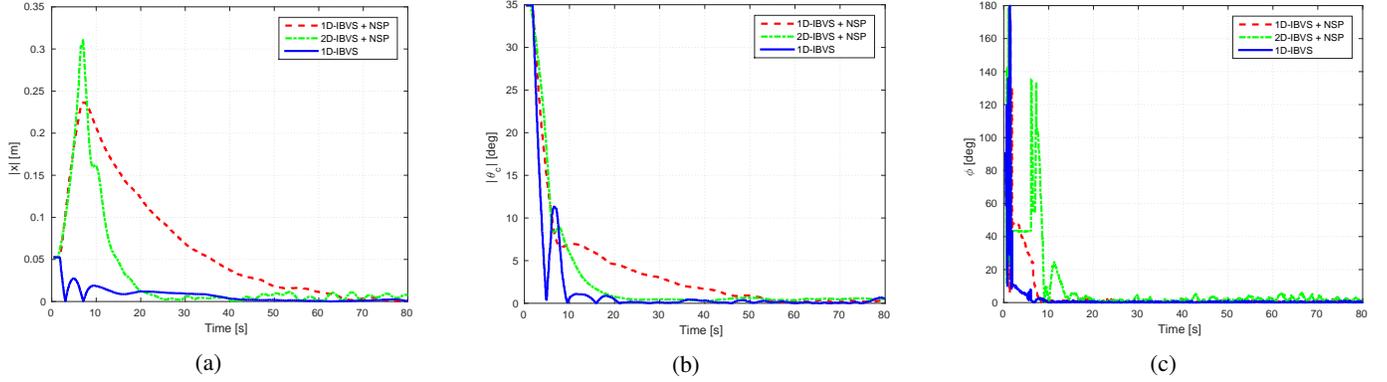


Fig. 4: Comparison between 1D IBVS, 1D IBVS with Null-Space Projection, and 2D IBVS with Null-Space Projection: (a) norm of the position error  $x$ , (b) absolute value of the orientation error  $\theta_c$  and (c) alignment error  $\phi$ .

Controller (6), in fact, relies on both the feedback provided by vision and on a nominal value of the robot velocity. In the following sections, we propose control schemes based only on a properly defined visual error and exhibiting different degrees of robustness with respect to velocity uncertainties.

### B. 1D IBVS with Null-Space projection

As done in Sect. III-A, we consider  $e_v = x_m$ , with dynamics given by (2). Since  $\mathbf{J}_x$  has dimension  $1 \times 3$ , the null-space  $\mathcal{N}(\mathbf{J}_x)$  has dimension 2. At steady state, when  $x_m = 0$ , the basis of  $\mathcal{N}$  is given by

$$\mathbf{B}_1 = \left\{ \begin{pmatrix} \cos q_p \\ \sin q_p \\ 0 \end{pmatrix}, \begin{pmatrix} -Z y_r / f \\ 0 \\ \sin q_p \end{pmatrix} \right\}, \quad (7)$$

from which we observe that the first vector of the basis is exactly the unit vector describing the direction of the principal axis of the camera in  $\mathcal{F}_r$ . Thus, the alignment constraint (4) is achieved by projecting in this null space a velocity vector satisfying (4). The control law has the form

$$\begin{aligned} \mathbf{v}_r &= -\mathbf{J}_x^\# (K_{x_m} x_m + J_\omega^x u_p) + (\mathbf{I}_3 - \mathbf{J}_x^\# \mathbf{J}_x) \mathbf{v}^* \\ u_p &= -K_p e_p \end{aligned} \quad (8)$$

being  $\mathbf{J}_x^\# = \mathbf{J}_x^T (\mathbf{J}_x \mathbf{J}_x^T)^{-1}$  the pseudo-inverse of  $\mathbf{J}_x$ ,  $\mathbf{I}_3$  the  $3 \times 3$  identity matrix, and  $\mathbf{v}^* = (v \cos q_p, v \sin q_p, 0)^T$ . Controller (8) guarantees the exponential convergence of  $x_m$  to 0 and, at the regulation point, satisfies constraint (4), through the projection of  $\mathbf{v}^*$  in the null-space (7).

We remark that (7) holds at the regulation point  $x_m = 0$ . During the transient, the basis of the null-space is not given by (7). As a result, the instantaneous direction of motion is not necessarily aligned with the camera principal axis, thus increasing the risk of collisions with undetected obstacles. In addition, controller (8) does not solve the problem of velocity uncertainties since regulation to zero of the feature  $x_m$  does not imply satisfaction of the constraint (4) by the robot actual velocity. In fact, at steady state, any velocity direction belonging to the space defined by (7) does not perturb the visual task. The next section shows that the definition of a 2D visual error allows to achieve the navigation task correctly.

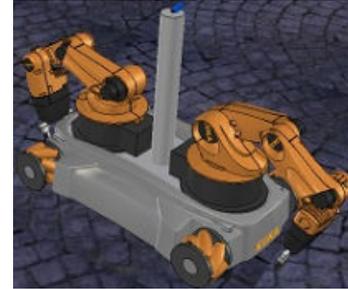


Fig. 5: The Youbot omnidirectional base with two manipulators and a monocular actuated camera on the top of the central grey cylinder.

### C. 2D IBVS with Null-Space Projection

In this case, the visual error is defined as  $e_v = (x_m, y_m)^T$ , with dynamics given by (2). The matrix  $\mathbf{J}$  has now dimension  $2 \times 3$ , thus the associated null-space  $\mathcal{N}(\mathbf{J})$ , far from singularities, has dimension 1. At  $x_m = 0$ , a basis for the null-space is

$$\mathbf{B}_2 = \left\{ \begin{pmatrix} \cos q_p \\ \sin q_p \\ 0 \end{pmatrix} \right\} \quad (9)$$

that is, as in the previous case, the unit vector describing the direction of the principal axis of the camera in  $\mathcal{F}_r$ . Thus, to guarantee the exponential convergence of  $e_v$  to  $\mathbf{0}$ , while keeping the linear velocity direction aligned with the camera focal axis, we design the following control law

$$\begin{aligned} \mathbf{v}_r &= -\mathbf{J}^\# (\mathbf{K} e_v + \mathbf{J}_\omega u_p) + (\mathbf{I}_3 - \mathbf{J}^\# \mathbf{J}) \mathbf{v}^* \\ u_p &= -K_p e_p \end{aligned} \quad (10)$$

with  $\mathbf{K} > 0$  and  $\mathbf{v}^* = (v \cos q_p, v \sin q_p, 0)^T$ . Differently from (8), the controller successfully aligns the motion direction with the camera principal axis if and only if  $e_v = \mathbf{0}$ . In fact, a velocity direction, not belonging to the null space  $\mathcal{N}(\mathbf{J})$ , i.e., not aligned with (9), will generate a visual error. However, during the transient  $\mathbf{B}_2$  is not given by (9), and a response similar to that generated by (8) will be observed.

### D. Controllers Comparison

To illustrate the behaviour of the three described controllers, this section proposes a set of preliminary simulations. The

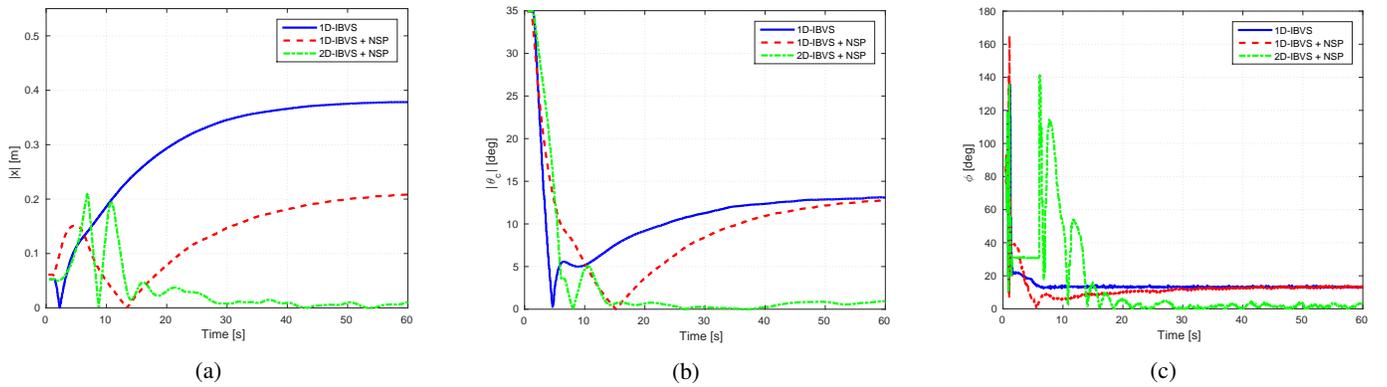


Fig. 6: Comparison between 1D IBVS, 1D IBVS with Null-Space Projection, and 2D IBVS with Null-Space Projection, in case of velocity uncertainties: (a) norm of the position error  $x$ , (b) absolute value of the orientation error  $\theta_c$  and (c) alignment error  $\phi$ .

simulated system is the mobile manipulator shown in Fig. 5 composed by the Youbot omnidirectional base carrying two manipulators and a monocular actuated camera positioned at the height of 45 cm and pointing downwards with tilt angle  $\gamma = 15^\circ$ . This setup is suitable for those applications where it is necessary to employ an omnidirectional robot with manipulation capabilities [11].

To evaluate the degree of violation of (4) we define the alignment error  $\phi = \cos^{-1}(\hat{\mathbf{v}}^T \hat{\mathbf{z}}_g)$ , with  $\hat{\mathbf{v}} = 1/v(v_x, v_y)^T$  the motion direction and  $\hat{\mathbf{z}}_g = (\cos q_p, \sin q_p)^T$  the ground projection of the camera focal axis.

The results shown in Fig. 4 compare the three controllers in terms of camera positioning error  $x$ , camera orientation error  $\theta_c$  (defined in Sect. II), and alignment error  $\phi$ . All the three controllers regulate the errors  $x$  and  $\theta_c$  to zero (Fig. 4a-4b). However, the transient duration and error are smaller for the 1D-IBVS, see Fig. 4c. This reduces the risk of collisions with undetected obstacles (see the accompanying video).

According to the above results, in nominal conditions, the 1D-IBVS control exhibits a safer transient behaviour with respect to the two alternative controllers. However, in the presence of actuation errors, due for example to modeling inaccuracies or mechanical faults, the actual robot velocity can differ from the nominal one, possibly violating (4). To analyse the effect of this error, we repeated the simulations considering additive perturbations on the nominal velocity components, i.e.,  $v_x = \bar{v}_x + \Delta v_x$ , and  $v_y = \bar{v}_y + \Delta v_y$ , where  $\bar{v}_{x,y}$  are the components of the commanded velocity, and  $\Delta v_{x,y}$  the additive, unknown, perturbation.

The simulation results obtained with  $\Delta v_x = 10\%v$  and  $\Delta v_y = 25\%v$  are shown in Fig. 6. The 1D-IBVS (6) and 1D-IBVS+NSP (8) are no longer able to regulate the camera pose along the corridor bisector. Instead, the 2D-IBVS+NSP (10) regulates the errors to zero. As explained in Sect. III-C, this occurs because the convergence of the visual error to zero is a necessary and sufficient condition for convergence of the camera to the corridor bisector in case of control (10). In fact, only velocities aligned with the camera gaze belong to the null space of the image Jacobian in (10) and, due to *Proposition 1*, this implies convergence of the camera pose to the corridor bisector. Nonetheless, the control strategy (10) is still unsafe during the transient response of the system,

as shown in Fig. 6c, where the violation of the alignment constraint indicates that it is not moving toward the direction it is looking at. In principle, this behaviour might be improved by fine tuning the controller gains. However, a robust choice of these parameters, based on a rigorous analysis of the robot dynamics is not easy, particularly for humanoids.

In the next Section, we provide an easier and cheaper solution. It extends (6) with an estimation of the robot velocity, used to close a loop on  $v_x$  and  $v_y$  to satisfy (4).

#### IV. 1D-IBVS CONTROLLER WITH VELOCITY FEEDBACK

This section illustrates the robot velocity regulation based on the velocity estimation provided by a Kalman filter.

##### A. Kalman filter design

The filter fuses data from IMU, encoders and monocular camera. Fusing proprioceptive data with visual information is a well-established approach in the literature [12] [13] [14]. In this case, we use the acceleration measured by the IMU in the filter prediction, to propagate an initial estimation of the robot velocity. Filter updates are obtained through the robot differential kinematics, relying on encoders data, and an estimation of the robot velocity based on optical flow [15] (as applied, e.g., in [16]). This last estimation comes at no additional computational cost since the optical flow is also computed to detect obstacles (see Sect. II).

We want to estimate the vector  $\xi = (\mathbf{v}_{IMU}^T, \omega_{IMU}^T)^T$  describing the 6D robot velocity expressed in the reference frame  $\mathcal{F}_{IMU}$  attached to the IMU of the robot. Since we use both visual (VIS) and differential-kinematics (DK) measurements for the filter update step, we assume a 12D measurement vector  $\mathbf{y}_k$ . Considering the accelerometer data  $\mathbf{a}_{IMU} = \mathbf{u}$  as input of the system, the discrete-time stochastic propagation model of the state vector  $\xi$  at step  $k$  is

$$\xi_{k+1} = \xi_k + \Delta T \begin{pmatrix} \mathbf{I}_3 \\ \mathbf{0}_3 \end{pmatrix} \mathbf{u}_k + \mathbf{n}_k, \quad (11)$$

where  $\Delta T$  is the sampling time and  $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{N}_k)$  is the zero-mean process noise with covariance matrix  $\mathbf{N}_k$ . The measurement model describing the output of the system is

$$\mathbf{y}_k = \begin{pmatrix} \mathbf{I}_6 \\ \mathbf{I}_6 \end{pmatrix} \xi_k + \mathbf{w}_k, \quad (12)$$

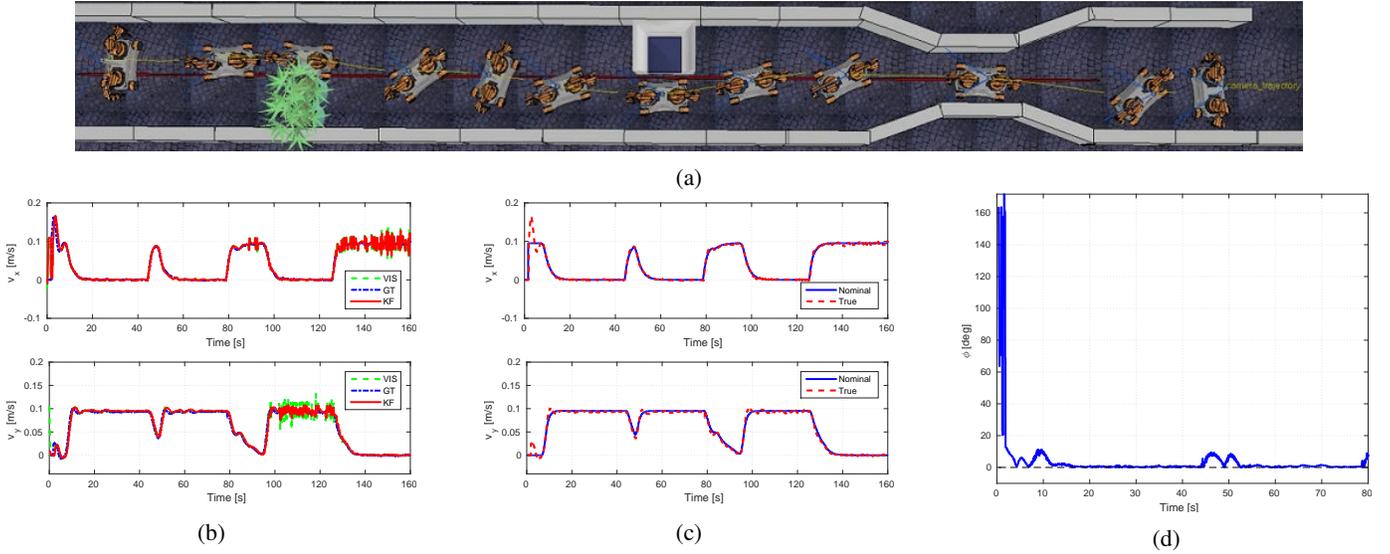


Fig. 7: (a) Omnidirectional wheeled robot moving among obstacles. Comparison of: (b) measured velocities: ground truth (GT) (blue), vision-based (green) and Kalman filter estimation (red); (c) reference velocity (blue) given by (4) and actual robot velocity (red), under the control (14). (d) Alignment error  $\phi$ .

where  $\mathbf{w}_k = (\mathbf{w}_{VIS}, \mathbf{w}_{DK})^T \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_k)$  is the zero-mean measurement noise with covariance matrix

$$\mathbf{W}_k = \begin{pmatrix} \mathbf{W}_{VIS,k} & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{W}_{DK,k} \end{pmatrix}, \quad (13)$$

with  $\mathbf{w}_{VIS,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_{VIS,k})$  and  $\mathbf{w}_{DK,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_{DK,k})$ . From (11) and (12), the standard equations of the Kalman filter are applied to estimate, at time  $t = k\Delta T$ ,  $\hat{\xi}_k$  and the associated covariance matrix  $\mathbf{P}_k$  [17].

### B. Velocity control loop

Once a reliable velocity estimation is available, we directly evaluate the error between a desired velocity  $\mathbf{v}_d$  (assigned according to (4)), and the estimation  $\mathbf{v}$  of the actual robot velocity provided by the filter. Given an acceleration input vector  $\mathbf{a}$ , the control is derived considering the ideal model for the center of mass of the robot  $\dot{\mathbf{v}} = \mathbf{a}$ . Then, defining the error  $\mathbf{e}_{xy} = \mathbf{v} - \mathbf{v}_d$ , to guarantee its exponential convergence to zero, we compute  $\mathbf{a} = \dot{\mathbf{v}}_d - \mathbf{K}_{xy}\mathbf{e}_{xy}$ , with  $\mathbf{K}_{xy} > 0$ . Integrating this acceleration input we get

$$\mathbf{v} = \mathbf{v}_d - \mathbf{K}_{xy} \int_0^t \mathbf{e}_{xy}(\tau) d\tau \quad (14)$$

which is the velocity commanded to the robot. This is an integral control of the robot velocity which does not require use of the (unavailable) robot position knowledge.

Used in combination with the 1D-IBVS control (6), this controller allows to always keep the robot velocity aligned with its gaze, thus solving the formulated navigation problem. In the next section, we show both simulation and experimental results validating the effectiveness of the controller.

## V. SIMULATIONS AND EXPERIMENTS

To validate the controller with velocity feedback, we performed simulations in the virtual environment V-REP, using an omnidirectional wheeled robot and the humanoid NAO. Two experiments were performed on a real NAO robot.

### A. Simulations with a wheeled robot

This group of simulations uses the omnidirectional mobile manipulator in Fig. 5, described in Sect. III-D. The camera image resolution is  $320 \times 240$ . A region of interest (ROI) of the source image has been considered for computation of the visual error to exclude detection of both too far and too close objects. This prevents the robot from reacting too fast and, at the same time, avoids that the robot body, when moving sideways, is erroneously considered as an obstacle. The robot velocity norm is  $v = 0.095$  m/s. We set  $d_{min} = \{165, 180\}$  pixels and  $d_{max} = \{185, 195\}$  pxl for the evaluation of  $q_p^*$ . For the estimation of the robot velocity, in this case, we only considered the vision-based velocity measurement (not the encoders). The covariance matrices considered for the estimation are  $\mathbf{N}_k = \text{diag}(10^{-10}, 10^{-10}, 10^{-10}, 10^{-1}, 10^{-1}, 10^{-1})$ ,  $\mathbf{W}_{VIS,k} = \text{diag}(10^{-9}, 10^{-9}, 10^{-8}, 10^{-8}, 10^{-8}, 10^{-8})$ . The control gains of the controller are  $K_{x_m} = 0.5$ ,  $K_p = 0.45$ ,  $\mathbf{K}_{xy} = \text{diag}(0.9, 0.9)$ . Finally, the velocity perturbation terms are set as  $\Delta v_x = 0.01$  m/s and  $\Delta v_y = 0.025$  m/s.

As navigation scenario, we considered a straight, long corridor with multiple obstacles generating narrow passages (see Fig. 7a). The figure shows the navigation pattern and the executed trajectory: when the robot is approaching a narrow passage, the control (6) re-orientes the robot by acting on the camera pan angle  $q_p$ , allowing the robot to safely cross the passage. In addition, control law (14) takes into account the actuation disturbances  $\Delta v_x$  and  $\Delta v_y$  by estimating the current robot navigation velocity (Fig. 7b), and correcting properly the commands to satisfy (4). Fig. 7c shows the tracking of the actual robot velocity w.r.t the desired values, while Fig. 7d shows the resulting alignment error  $\phi$ .

### B. Simulations with a humanoid robot

For the simulations with a humanoid robot, we used the small robot NAO. The robot is equipped with a pair of monocular cameras, mounted on top and on bottom of the

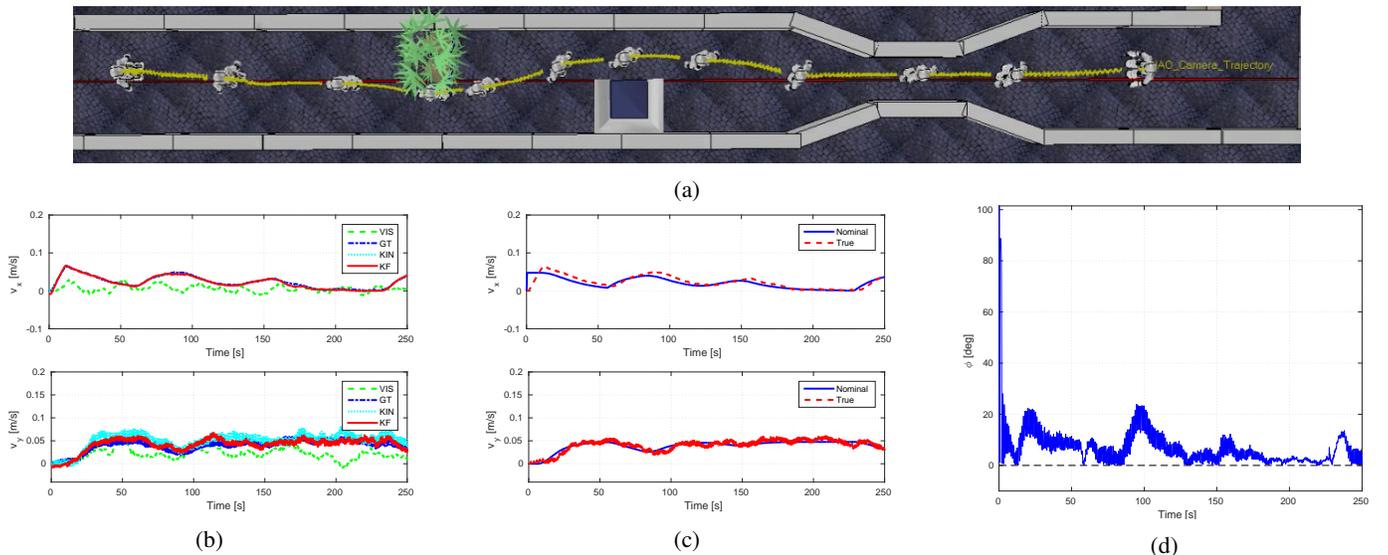


Fig. 8: (a) NAO moving among obstacles. Comparison of: (b) measured velocities: ground truth (blue), vision-based (green), kinematics-based (cyan) and Kalman filter estimate (red); (c) reference velocity (blue) given by (4) and actual velocity (red), with control (14). (d) Alignment error  $\phi$ .

head. To avoid undesired image occlusions caused by the robot body, we only used the top camera. The robot is commanded through the built-in library NAOqi, that allows to directly command the robot translational velocity components  $v_x$ ,  $v_y$  and the angular velocity  $\omega$ . The main issue in implementing the proposed control scheme on a humanoid robot lies in the *sway motion* caused by the robot walk. This may have drastic effects on vision-based measurement processes involved in the navigation scheme, since the optical flow is prone to noise due to image blur. While it is possible to identify and isolate the sway motion from the overall motion of the robot [18] [19], satisfying results are also achieved by filtering out the generated oscillations in the affected signals [19]. Inspired by this latter work, we applied a low-pass filter on the detected dominant plane and the sensor measurements used for robot velocity estimation. In addition, the oscillations also generate erroneous acceleration measurements, due to the projection of the gravity vector  $\mathbf{g} = [0, 0, g]^T$  along the  $x$ - and  $y$ -axis of  $\mathcal{F}_{IMU}$ . We compensate such projections to generate reliable data for the Kalman filter. By measuring the torso angles  $\alpha$  and  $\beta$  w.r.t.  $\mathbf{g}$  through NAOqi, the compensated acceleration  $\mathbf{a}$  is computed from the raw measurement  $\hat{\mathbf{a}}$  as  $\mathbf{a} = \hat{\mathbf{a}} - \mathbf{g}\mathbf{b}$ , where  $\mathbf{b} = (\sin \beta, \cos \beta \sin \alpha, \cos \beta \cos \alpha)^T$ . We set  $\gamma = 11.45^\circ$ ,  $v = 0.0476$  m/s,  $d_{min} = \{100, 133\}$  and  $d_{max} = \{140, 145\}$  pxl. For the robot velocity estimation, we used filtered vision- and kinematics-based velocity measurements in the filter update, with covariances  $\mathbf{W}_{VIS,k} = \text{diag}(10^{-6}, 10^{-6}, 10^{-6}, 10^{-7}, 10^{-7}, 10^{-7})$  and  $\mathbf{W}_{KIN,k} = \text{diag}(10^{-9}, 10^{-9}, 10^{-9}, 10^{-8}, 10^{-8}, 10^{-8})$ . The control gains are  $\mathbf{K}_{xy} = \text{diag}(0.04, 0.04)$ ,  $K_{x_m} = 0.4$ ,  $K_p = 0.05$ . As actuation disturbance, we set  $\Delta v_x = 10\%v$ .

We validated the method with two simulations. First, we considered a straight corridor with multiple obstacles and narrow passages. Fig. 8a shows the trajectory and the reconfigurations of the robot: the controller compensates the actuation disturbance estimated through the Kalman filter (Fig. 8b)

and allows to track the reference velocity determined by (4) (Fig 8c). The alignment error  $\phi$  is shown in Fig. 8d. The peak values of the error correspond to changes on the robot body orientation to cross narrow passages. In the second simulation, shown in the accompanying video, NAO walks in a general cluttered environment, avoiding obstacles and moving sideways to cross a narrow passage.

### C. Experiments with the humanoid robot

Experiments to validate the proposed control framework have been performed with the robot NAO. Velocity ground truth is provided by a VICON Motion Capture System. Three markers have been placed on the robot head to identify the reference frame  $\mathcal{F}_m$ . The homogeneous transformation matrix  ${}^m\mathbf{T}_c$  between  $\mathcal{F}_m$  and  $\mathcal{F}_c$  has been estimated through a least-square-based camera-VICON calibration algorithm.

The scenario consists of a straight corridor with a narrow passage, made of wooden panels and a patched carpet. The injected velocity perturbation term is  $\Delta v_x = 20\%v$ . Fig. 9 shows NAO navigating with and without velocity feedback control: on top, the robot walks under controls (4)-(6), but (4) is violated due to the actuation disturbance, since the actual navigation velocity is not taken into account. The velocity is not aligned with the camera gaze and the robot collides soon against one of the corridor walls. On bottom, using the control (14) the robot velocity converges to the reference values satisfying (4). Thus, the alignment error  $\phi$  converges to 0 (Fig 10) and the robot crosses the narrow passage without collisions, recovering the corridor center.

## VI. CONCLUSIONS

Through a comparative analysis of three vision-based control strategies for navigation of omnidirectional robots in environments with obstacles, this work shows that a single, appropriate, feature and gaze-velocity alignment guarantee obstacle avoidance. The controllers share the characteristic

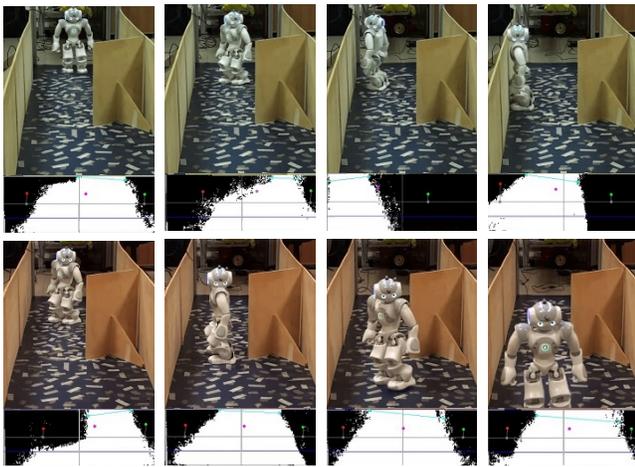


Fig. 9: NAO navigating in a corridor with a narrow passage, and a velocity perturbation of  $\Delta v_x = 20\%v$ . Navigation under the control (4)-(6) (top) and (6)-(14) (bottom). For each snapshot, the *dominant plane* is also shown.

of aligning, at steady state, the camera gaze with the robot velocity, thus allowing the robot to move in the direction where it is looking while orienting the body in accordance to the size of the traversable space.

We show that the direct enforcement of the gaze-velocity constraint minimises the risk of collisions also during the transient phases of navigation. To robustly enforce the gaze-velocity constraint the actual robot velocity is estimated and fed back to the controller.

Future work includes analysis of the method performance in the presence of moving obstacles.

## REFERENCES

- [1] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, 2006.
- [2] A. Khelloufi, N. Achour, R. Passama, and A. Cherubini, "Tentacle-based moving obstacle avoidance for omnidirectional robots with visibility constraints," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017.
- [3] C. Sprunk, B. Lau, P. Pfaff, and W. Burgard, "An accurate and efficient navigation system for omnidirectional robots in industrial environments," *Autonomous Robots*, vol. 41, no. 2, 2017.
- [4] A. Schaub, D. Baumgartner, and D. Burschka, "Reactive obstacle avoidance for highly maneuverable vehicles based on a two-stage optical flow clustering," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, 2017.
- [5] O. Stasse, "SLAM and Vision-based Humanoid Navigation," in *Humanoid Robots: A Reference*, A. Goswami and P. Vadakkepat, Eds. Springer, 2018, ch. Self-localization and Map building.
- [6] D. Maier, C. Stachniss, and M. Bennewitz, "Vision-based humanoid navigation using self-supervised obstacle detection," *Int. Journal of Humanoid Robotics*, vol. 10, no. 2, 2013.
- [7] A. Paolillo, A. Faragasso, G. Oriolo, and M. Vendittelli, "Vision-based maze navigation for humanoid robots," *Autonomous Robots*, vol. 41, no. 2, 2017.
- [8] J. Delfin, H. M. Becerra, and G. Arechavaleta, "Humanoid navigation using a visual memory with obstacle avoidance," *Robotics and Autonomous Systems*, vol. 109, 2018.
- [9] M. Ferro, A. Paolillo, A. Cherubini, and M. Vendittelli, "Omnidirectional humanoid navigation in cluttered environments based on optical flow information," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2016.
- [10] N. Ohnishi and A. Imiya, "Appearance-based navigation and homing for autonomous mobile robot," *Image and Vision Computing*, vol. 31, no. 6, 2013.
- [11] B. Navarro, A. Cherubini, A. Fonte, G. Poisson, and P. Fraisse, "A framework for intuitive collaboration with a mobile manipulator," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017.

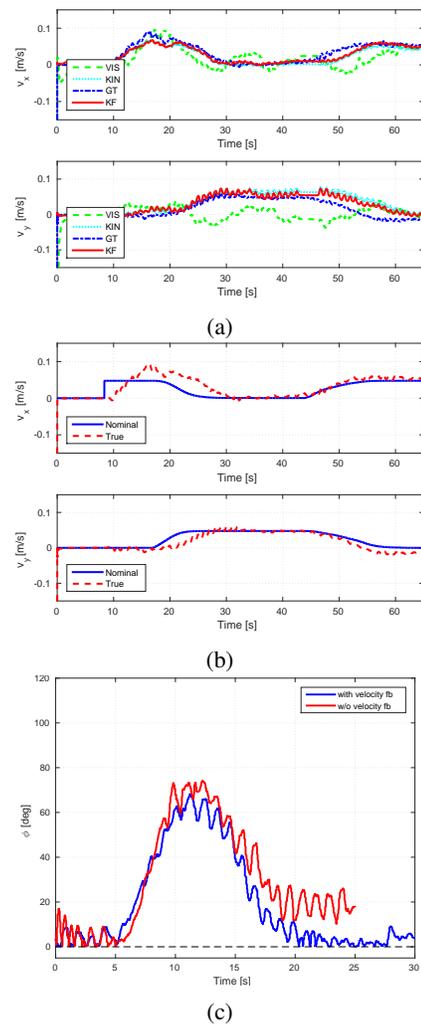


Fig. 10: Experimental results. Comparison of: (a) measured velocities: VICON ground truth (GT) (blue), vision-based (magenta), kinematics-based (green) and Kalman filter estimation (red); (b) reference velocity (red) given by (4) and actual velocity (blue), using the control (6)-(14); (c) errors  $\phi$ , between the experiments of Fig. 9.

- [12] V. Grabe, H. H. Bühlhoff, D. Scaramuzza, and P. R. Giordano, "Nonlinear ego-motion estimation from optical flow for online control of a quadrotor UAV," *The International Journal of Robotics Research*, vol. 34, no. 8, 2015.
- [13] G. Oriolo, A. Paolillo, L. Rosa, and M. Vendittelli, "Humanoid odometric localization integrating kinematic, inertial and visual information," *Autonomous Robots*, vol. 40, no. 5, 2016.
- [14] K. McGuire, G. de Croon, C. De Wagter, K. Tuyls, and H. J. Kappen, "Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, 2017.
- [15] R. Barbosa, J. Silva, M. M. Junior, and R. Gallis, "Velocity estimation of a mobile mapping vehicle using filtered monocular optical flow," in *Int. Symp. on Mobile Mapping Technology*, vol. 5, 2007.
- [16] A. Paolillo, P. Gergondet, A. Cherubini, M. Vendittelli, and A. Kheddar, "Autonomous car driving by a humanoid robot," *Journal of Field Robotics*, vol. 35, no. 2, 2018.
- [17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
- [18] C. Dune, A. Herdt, O. Stasse, P.-B. Wieber, K. Yokoi, and E. Yoshida, "Cancelling the sway motion of dynamic walking in visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010.
- [19] G. Oriolo, A. Paolillo, L. Rosa, and M. Vendittelli, "Vision-based trajectory control for humanoid navigation," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2013.