Available online at www.sciencedirect.com
ScienceDirect
Robotics and Autonomous Systems 55 (2007) 419–432
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Vision-based interception of a moving target with a nonholonomic mobile robot

Luigi Freda, Giuseppe Oriolo*

Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Eudossiana 18, 00184 Roma, Italy

Received 21 January 2006; received in revised form 26 January 2007; accepted 2 February 2007
Available online 13 February 2007

Abstract

A novel vision-based scheme is presented for driving a nonholonomic mobile robot to intercept a moving target. The proposed method has a two-level structure. On the lower level, the pan–tilt platform carrying the on-board camera is controlled so as to keep the target as close as possible to the center of the image plane. On the higher level, the relative position of the target is retrieved from its image coordinates and the camera pan–tilt angles through simple geometry, and used to compute a control law which drives the robot to the target. Various possible choices are discussed for the high-level robot controller, and the associated stability properties are rigorously analysed. The proposed visual interception method is validated through simulations as well as experiments on the mobile robot MagellanPro.

Keywords: Visual servoing; Target interception; Nonholonomic mobile robots; Mobile robot control; Camera tracking

1. Introduction

This paper deals with the problem of intercepting a moving target via a nonholonomic mobile robot through visual feedback. Interception (approaching a moving object until collision) and tracking (approaching a moving object while matching its location and velocity) are important tasks in a number of applications, ranging from robotic games to automated surveillance. Moreover, the development of effective methods for performing these tasks represents a challenging testbed for the integration of various techniques involving image processing, filtering, control theory and artificial intelligence (AI) strategies.

Many different approaches have been proposed in the literature for interception and tracking, depending on the target motion characteristics as well as the robot kinematic and dynamic model. In principle, two opposite situations may occur: the instantaneous target location and velocity can be known in advance as part of a reference trajectory, or it can be estimated and predicted through sensory (typically visual) data.

When the target motion is completely (or to a large extent) known in advance, time-optimal interception can be attempted. In this case, the problem becomes essentially that of planning a robot trajectory leading to a convenient rendezvous point. The general strategy known as PPE (Prediction, Planning and Execution) relies on this approach. For example, a time-optimal technique for free-flying interceptors and targets is proposed in [24]. This kind of techniques, combined with conventional tracking methods, can be extended to manipulators so as to achieve smooth grasping interception with terminal-velocity matching [17]. In [10], an active PPE technique for a six-degrees-of-freedom (6-dof) manipulator is introduced in order to improve the basic PPE strategy.

When the target motion is not known in advance and cannot be safely predicted (for example, because sudden changes may occur), it is necessary to rely on some form of estimation of its position and velocity on the basis of sensor data. In this sense, visual feedback [4] is an appealing possibility. In *position-based* visual servoing, the target posture is estimated on the basis of visual data and geometric models. For instance, an omnidirectional vision system is used in [11] to determine the robot posture from extracted features, so that the basic tasks of a robotic goalkeeper can be accomplished in terms of trajectory tracking and posture stabilization. In this context, Kalman filtering is often used to obtain a robust prediction of the target motion [1,7,16].

* Corresponding author. Tel.: +39 06 4458 5874; fax: +39 06 44584367.
E-mail addresses: freda@dis.uniroma1.it (L. Freda), oriolo@dis.uniroma1.it (G. Oriolo).

0921-8890/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2007.02.001

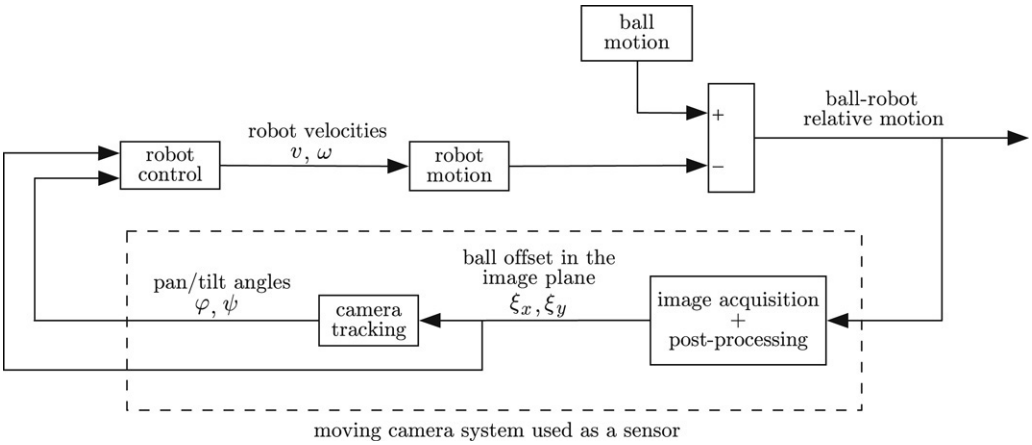

Fig. 1. A block diagram description of the proposed approach.

In *image-based* visual servoing, the spatial relationship between the target and the robot camera is directly estimated on the image plane, and the error signal is expressed in terms of image parameters. This approach, originally developed for robotic manipulators equipped with an eye-in-hand system, has also been extended to nonholonomic vehicles such as wheeled mobile robots [15,23].

Many works address trajectory tracking directly on the image plane. An image-based motion planning scheme is proposed in [25], where a virtual trajectory is directly generated and tracked on the image plane. In [6], a mobile robot tracks road edges in a panoramic image sequence. In [14], a wheeled vehicle tracks a trajectory represented in terms of image sequences of an object, and the controlled camera motion is used as an input for visual servoing. A similar concept was used in [5]: while the camera tracks the ball on the image plane, pan and tilt angles are used as a visual input for a probabilistic motion control scheme.

Finally, another approach to the interception problem with mobile robots is constituted by *human-like strategies* like the so-called LOT and OAC [2,18]. These works develop some heuristics, like the optical acceleration cancellation method, in order to accomplish the interception of free-flying objects or targets moving on the ground [20]. In spite of the simplicity and interest of these strategies, no analytical proofs have been provided for the stability (and hence the convergence) of the associated controllers.

Our approach to the interception problem is to design a control scheme which makes use of visual information only and does not need long-term predictions of the target motion. This last assumption is particularly motivated by the potential application in the general case in which the target motion is unknown in advance and may be characterized by sudden changes or reversals (e.g., robot soccer). The nonholonomic nature of the mobile robot considered in this paper is an additional (but realistic) complication, because the associated control problem becomes considerably more difficult with respect to the case of free-flying interceptors.

The proposed control scheme has a two-level structure (see Fig. 1). On the lower level, the pan–tilt platform which carries the on-board camera is controlled so as to keep the target close to the center of the image plane. On the higher level, the relative position of the ball is retrieved from its image coordinates and the camera pan–tilt angles through simple geometry, and used to compute a control law driving the robot to the target. This scheme implicitly assumes a *separation* between the pan–tilt platform dynamics and the robot closed-loop dynamics which is reminiscent of observer-based stabilization schemes.

The paper is organized as follows. A general description of the method and a statement of the working assumptions is given in Section 2. The adopted target recognition algorithm and the low-level camera control laws are presented in Section 3. Then, various possible choices for the high-level robot controller (including two original feedback laws) and their visual implementations are discussed in Section 4. To validate the overall visual interception method, and especially the assumption of separation between the camera and robot dynamics, simulations as well as experiments on the mobile robot MagellanPro are presented in Section 5. In the concluding section, possible extensions of the method are discussed. Analytical proofs of stability of the high-level robot controllers are collected in the Appendix.

2. General description of the approach

The objective of this paper is to devise a vision-based control method for driving a wheeled mobile robot to intercept a moving target. In particular, we shall take the following assumptions:

1. The target and the robot move on the same plane in the absence of obstacles.
2. The moving target is a ball whose trajectory is generic and unknown in advance.
3. The robot is subject to a nonholonomic rolling constraint. In particular, it has the kinematics of a unicycle:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{1}$$

where x, y are the robot Cartesian coordinates in a fixed frame, θ is its orientation with respect to the same frame, and v and ω are the robot linear and angular velocities.

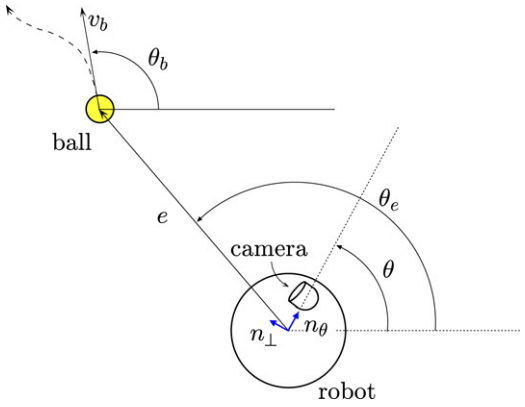

Fig. 2. Problem setting with definition of some relevant variables.

4. The robot is equipped with an on-board camera mounted on a pan–tilt platform.

The generality of the second assumption is motivated by applications where the target frequently changes speed and direction. As a consequence, unlike other proposed techniques, our visual interception approach does not make *any* prediction of the ball movement. All the control laws to be presented only use the ball position and velocity as estimated from visual data.

We propose a two-level solution whose conceptual scheme is outlined in Fig. 1, with reference to the problem setting illustrated in Fig. 2. The basic idea is that on the lower level the pan–tilt platform carrying the camera system is controlled so as to keep the ball at the center of the image. During this process, the acquired images are post-processed in order to retrieve the coordinates of the ball center w.r.t. the center of the image plane (the *ball offset*), and then a camera tracking controller moves the pan–tilt platform so as to zero the offset. On the higher level, the relative position of the ball with respect to the robot can be retrieved through simple geometry from its image offset and from the pan–tilt angles. This reconstructed information is then used to drive the robot toward the ball. In practice, the whole pan–tilt camera system acts as a sensor, providing information about the relative motion of the target with respect to the ball.

The above strategy is reminiscent of a human-like behavior: the body follows the head while the latter visually pursues a target. By following this approach, we are implicitly assuming a separation between the camera and the robot closed-loop dynamics (very similar to the property underlying observer-based stabilization schemes). This assumption appears reasonable also considering that the pan–tilt platform dynamics is typically much faster than the robot dynamics. Note also that the robot odometry is never used; as a consequence, the controllers to be presented in the following are purely image-based.

For illustration, our method will be discussed with reference to a MagellanPro robot. This is a differential-drive robot with a caster wheel added for stability; its shape is circular with a diameter of 40 cm. The on-board camera is a Sony EVI-D31, with a focal length of 5.4 mm, a refresh time of 0.13 s and a resolution of 160 × 120 pixels. The camera is mounted on a pan–tilt platform which provides two degrees of freedom. For simplicity, we shall neglect the small displacement (about 11 mm) between the pan and the tilt axes.

3. Image processing and camera tracking

The camera tracking algorithm must keep the ball as close as possible to the center of the image plane. To this end, it is first necessary to recognize the ball in the image on the basis of its known color and shape.

3.1. Image processing

As a first step, color recognition is used to obtain a binary image. This process is executed on the basis of a ball color histogram computed during a preliminary color calibration.

Next, to reduce noise, the binary image is processed through two consecutive steps [21]: an erosion and then a dilation are applied to 'clean' the image by respectively eliminating spurious pixels outside and inside the reconstructed ball blob. The result of the first two steps is shown in Fig. 3.

If in the environments there are other objects with the same color of ball, shape recognition based on Hu-moments [9] is also performed. The result of shape recognition is shown in Fig. 4.

3.2. Camera tracking

Different visual servoing approaches are available in the literature for tracking a moving target with a mobile camera. The basic approach relies on the use of the image Jacobian which relates, in terms of differential changes, the image feature parameters to the camera position coordinates [4]. Other available methods estimate this relationship through different techniques, e.g., using a probabilistic framework [5] or a fuzzy

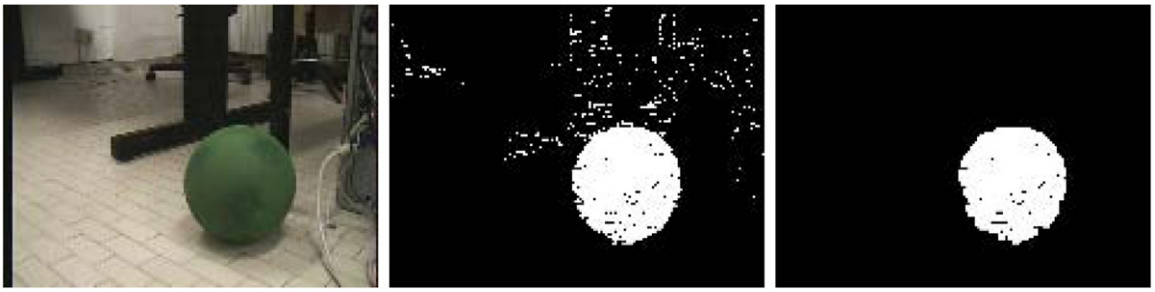

Fig. 3. *Left*: The image acquired by the camera. *Center*: After color recognition. *Right*: After noise filtering.

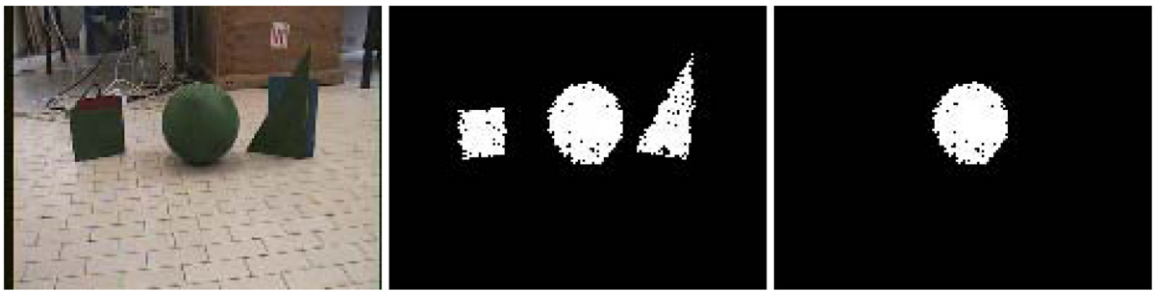

Fig. 4. *Left*: Three objects with the same color. *Center*: After color recognition and filtering. *Right*: After shape recognition.

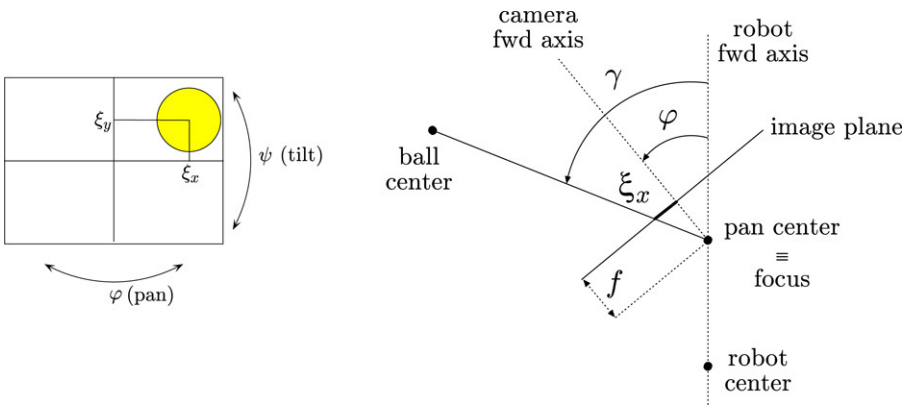

Fig. 5. *Left*: definition of the ball offset. *Right*: dependence of the ball image coordinate ξ_x on the pan angle φ.

logic approach [12]. Our choice was to devise a camera control scheme which relies on a simple geometric model and which can be easily integrated in the overall two-level interception scheme, allowing the pan–tilt camera to independently track the moving target.

Let $\xi = (\xi_x\ \xi_y)^{\mathrm{T}}$ be the ball offset (i.e., the vector of the ball coordinates w.r.t. the center of the image) computed from the image as the coordinates of the centroid of the ball (see Fig. 5, left). For the design of the camera tracking method, we make the simplifying assumption that, for a given position of the ball, ξ_x and ξ_y on the image plane respectively depend only on the pan angle φ and on the tilt angle ψ. Under such approximation, the expression of these relationships can be computed by planar geometry.

For illustration, consider the dependence of ξ_x on the pan angle φ. Refer to Fig. 5, right, for the geometric setting and definitions; note that the pan center (i.e., the point where the pan axis intersects the camera axis) is assumed to coincide with the focus of the camera.[1] Let f be the camera focal length. The following equation holds:

$$\xi_x = f \tan(\gamma - \varphi)$$

so that the differential kinematics of ξ_x is

$$\dot{\xi}_x = \frac{f}{\cos^2(\gamma - \varphi)}(\dot{\gamma} - \dot{\varphi}).$$

[1] One may verify that such a simplification is acceptable in practice, provided that the displacement between these two points (less than 3 cm for the MagellanPro) is much smaller than the distance between the ball center and the pan axis.

This represents the kinematic model upon which we will design the camera tracking controller for the pan angle. The pan velocity $\dot{\varphi}$ is considered as an available input and $\dot{\gamma}$ represents an exogenous input because it depends on the instantaneous relative motion between the ball and the robot.

Choosing the pan velocity as

$$\dot{\varphi} = \dot{\gamma} + k_\varphi \xi_x \quad k_\varphi > 0 \tag{2}$$

guarantees exponential convergence of ξ_x to zero. In fact, the closed-loop dynamics is

$$\dot{\xi}_x = -\frac{f k_\varphi}{\cos^2(\gamma - \varphi)}\xi_x$$

with the modulus of the time-varying gain certainly larger than $f k_\varphi$.

The feedforward term $\dot{\gamma}$ can be computed numerically on the basis of the following relationship:

$$\gamma = \varphi + \arctan \xi_x / f. \tag{3}$$

The addition of this feedforward term compensates for any relative motion between the ball and the robot, thus increasing the level of separation between the camera and the robot dynamics.

A similar construction in the vertical plane holds for the dependence of the other ball coordinate ξ_y on the tilt angle ψ. Hence, choosing the tilt velocity as

$$\dot{\psi} = \dot{\eta} + k_\psi \xi_y \quad k_\psi > 0 \tag{4}$$

guarantees exponential convergence of ξ_y to zero. Here, η is the angle between the horizontal axis and the line joining the

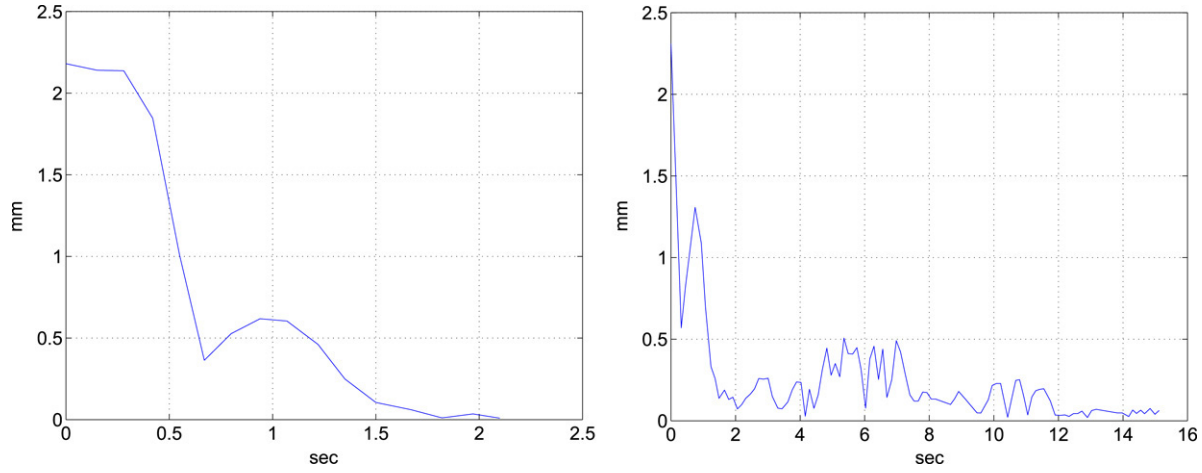

Fig. 6. Experimental results of camera tracking in terms of the norm $\|\xi\|$ of the ball offset. *Left*: the robot and the ball are motionless. *Right*: the robot is moving along a straight path at a speed of 0.2 m/s.

camera focus and the ball center, computed as

$$\eta = \psi + \arctan \xi_y / f. \tag{5}$$

Again, the feedforward $\dot{\eta}$ is computed via finite difference.

3.3. Tracking experiments

Before proceeding with the description of the complete control scheme, we would like to assess the performance of the proposed camera tracking method in itself. Fig. 6 shows its performance in two experiments executed with the actual MagellanPro robot. In the first experiment, the robot and the ball are motionless; the camera successfully hunts the ball and places it at the center of the image, zeroing the ball offset $\|\xi\|$. In the second experiment, the robot is moving along a straight path at a speed of 0.2 m/s, while the ball is still and there is an initial offset. The residual norm of the ball offset is very small and essentially due to the quantization introduced by the camera. These satisfactory results justify *a posteriori* the simplifying assumption that ξ_x and ξ_y respectively depend only on φ and ψ.

4. Robot control

In this section we describe the three different robot controllers that have been integrated in our interception scheme. Of these, the first two are original, while the third is a well-known tracking method.

The control laws are first presented in a *complete information* context: both the system (the robot) state and the *exosystem* (the ball) state are assumed to be known. Then, it is shown how these control laws can be implemented knowing only the pan–tilt angles, based on the assumption that the camera tracking system effectively keeps the ball at the center of the image.

4.1. Structure of the controllers

Refer again to Fig. 2 for the geometry of the problem. Let e be the cartesian error vector, v_b the ball velocity vector, and n_θ and $n_\perp$ the unit vectors respectively parallel and orthogonal to the robot forward axis. Moreover, denote by θ_b the absolute orientation of v_b, and by θ_e the absolute orientation of e. In the following, we will refer to $\theta_e - \theta$ as the *pointing error* and to $\theta_b - \theta$ as the *alignment error*. The former represents the angle between the robot forward axis and the line of sight to the target; the latter is the angle between the instantaneous directions of motion of the target and the ball.

The first controller, inspired by the Cartesian position regulator of [8], is expressed as

$$\begin{aligned} v &= k_1\, e^{\mathrm{T}} n_\theta + v_b^{\mathrm{T}} n_\theta \\ \omega &= k_2(\theta_e - \theta) + \dot{\theta}_e \end{aligned} \tag{6}$$

with $k_i > 0$, $\forall i$. The linear velocity is composed by a feedback term and a feedforward term, respectively obtained by projecting on the forward robot axis the error vector and the ball velocity vector. The angular velocity includes a feedback action proportional to the pointing error and a feedforward term given by the derivative of the angle θ_e. For the above control law, a Lyapunov-like analysis allows us to establish global asymptotic stability of the cartesian and pointing error (i.e., e and $\theta_e - \theta$) at the origin, under the assumption that the target itself moves as a unicycle (see the Appendix, Proposition 1).

A second controller may be derived from the first by adding to the angular velocity the feedback term $e^{\mathrm{T}} n_\perp$, i.e., the projection of the error vector along the orthogonal direction to the forward axis of the robot:

$$\begin{aligned} v &= k_1\, e^{\mathrm{T}} n_\theta + v_b^{\mathrm{T}} n_\theta \\ \omega &= k_2(\theta_e - \theta) + k_3\, e^{\mathrm{T}} n_\perp + \dot{\theta}_e \end{aligned} \tag{7}$$

with $k_i > 0$, $\forall i$. This additional feedback term forces a faster alignment of the robot with the line of sight of the target when the norm of the vector error is large, i.e., when the robot is far from the ball. This should result in a more effective interception of the target. Also for this controller, global asymptotic stability of the cartesian and pointing error dynamics at zero can be proved (see the Appendix, Proposition 2) when the target moves

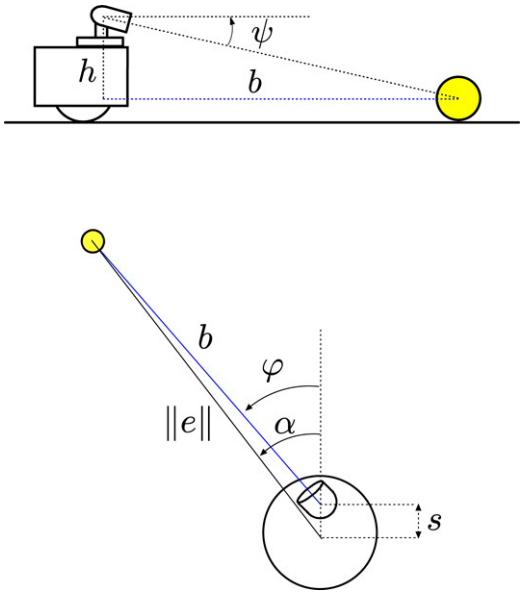

Fig. 7. Determination of the ball position with respect to the robot from the pan–tilt angles under the assumption of perfect camera tracking. *Above*: Side view. *Below*: Top view.

as a unicycle. In particular, the speed of convergence with this controller is always higher than with (6) (Appendix, Remark 1).

The third control law chosen for our experimental study is a well-known trajectory tracking method [22] for nonholonomic robots. This controller is quite similar to the previous ones, the difference being the use in the angular velocity of the alignment error $\theta_b - \theta$ in place of the pointing error $\theta_e - \theta$, and of a nonlinear gain for the second feedback term:

$$\begin{aligned} v &= k_1\, e^{\mathrm{T}} n_\theta + v_b^{\mathrm{T}} n_\theta \\ \omega &= k_2(\theta_b - \theta) + k_3 |v_b| \frac{\sin(\theta_b - \theta)}{\theta_b - \theta}\, e^{\mathrm{T}} n_\perp + \dot{\theta}_b \end{aligned} \tag{8}$$

with $k_i > 0$, $\forall i$. Note that, consistently, the feedforward term in ω is different as well. In this case the robot is guaranteed to converge globally in position and orientation to the ball, i.e., e and $\theta_b - \theta$ converge to zero [22]. In this sense, with respect to the previous schemes, controller (8) aims more at tracking than at intercepting the target.

It is interesting to note that all three controllers prescribe the same linear velocity for the robot. One basic difference among the first two and the third is the following: while controller (8) requires that the target does not stop,[2] controllers (6) and (7) will guarantee interception also in that case, due to their primary nature of position (as opposed to posture) stabilizers. In this regard, see the Appendix, Remark 2.

4.2. Visual implementation of the controllers

In this section, we show how all the variables needed to implement the interception controllers (6)–(8) can be computed from the pan–tilt angles and the ball offset in the image plane. In the following, we will often collectively refer to these computations as *ball localization*.

For the sake of illustration, we shall first make the assumption of perfect camera tracking (i.e., the ball is exactly at the center of the image plane), showing that ball localization in this particular case can be performed using only the pan–tilt angles. We will then remove such assumption and consider the general case.

Refer to Fig. 7 for the relevant geometry and the definition of auxiliary variables. Let b and h be respectively the horizontal and vertical displacement between the tilt axis and the ball center. The following equation holds:

$$b = \frac{h}{\tan\psi}. \tag{9}$$

The application of Carnot's theorem gives

$$\|e\|^2 = b^2 + s^2 - 2bs\cos(\pi - \varphi) \tag{10}$$

where s is the horizontal displacement between the robot center and the pan axis. The angle α between the error vector and the forward axis of the robot is

$$\alpha = \arcsin \frac{b\sin(\pi - \varphi)}{\|e\|}. \tag{11}$$

From these variables it is possible to compute the pointing error used by controllers (6) and (7) as

$$\theta_e - \theta = \alpha$$

while the projections of the error vector along the robot forward axis and its orthogonal direction are, respectively,

$$e^{\mathrm{T}} n_\theta = \|e\|\cos\alpha \qquad e^{\mathrm{T}} n_\perp = \|e\|\sin\alpha.$$

The first term appears in all three controllers, while the second only enters in controllers (7) and (8).

The remaining quantities needed for control computation can be approximated through finite differences as follows. Fig. 8 shows the ball in two positions P_1 and P_2 attained at consecutive instants t_1 and t_2 separated by a small sampling interval T. A backward approximation of the ball velocity vector at t_2 can be computed as $v_b \approx \overrightarrow{P_1P_2}/T$. Assuming the frame attached to the robot in t_1 to be a reference frame, with the x axis along n_{θ_1}, the incremental robot displacement can be estimated by the Runge–Kutta second-order method as

$$\Delta x \approx v_1 T\cos\frac{\omega_1 T}{2} \qquad \Delta y \approx v_1 T\sin\frac{\omega_1 T}{2} \qquad \Delta\theta \approx \omega T.$$

In the following, the subscript i ($i = 1, 2$) will refer to the time instant in which each quantity is defined. Note, in particular, that v_1 and ω_1 are known because they are the linear and angular velocity inputs to the robot at the previous sampling instant.

The components of $\overrightarrow{P_1P_2}$ can be computed as follows:

$$(\overrightarrow{P_1P_2})_x \approx v_1 T\cos\frac{\omega_1 T}{2} + \|e_2\|\cos(\alpha_2 + \omega_1 T) - \|e_1\|\cos\alpha_1$$

$$(\overrightarrow{P_1P_2})_y \approx v_1 T\sin\frac{\omega_1 T}{2} + \|e_2\|\sin(\alpha_2 + \omega_1 T) - \|e_1\|\sin\alpha_1.$$

[2] It may be verified that when the target stops, controller (8) does not guarantee convergence of the robot position error to zero (see also the last simulation in Section 5.1).

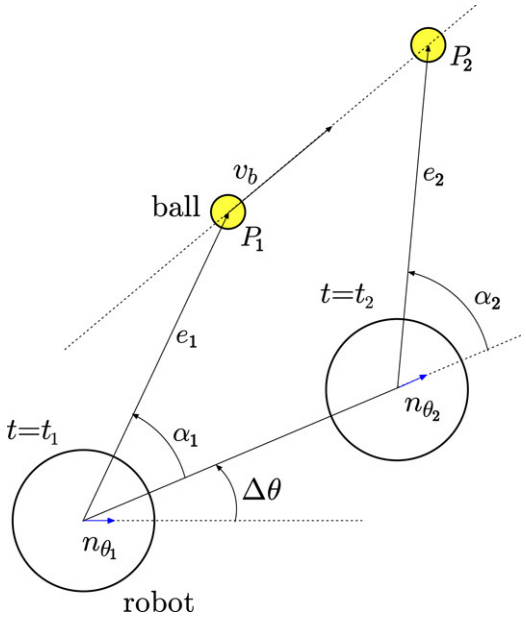

Fig. 8. Numerical estimation of the ball velocity.

An estimate of the alignment error at t_2 is then given by

$$\theta_b - \theta \approx \text{ATAN2}((\overrightarrow{P_1P_2})_y, (\overrightarrow{P_1P_2})_x) - \omega_1 T.$$

This allows us to compute the feedback part in the angular velocity of controller (8).

An easy calculation yields the estimate of the linear velocity feedforward term of the three controllers:

$$v_b^{\mathrm{T}} n_\theta \approx v_1 \cos\frac{\omega_1 T}{2} + \frac{\|e_2\|\cos\alpha_2}{T} - \frac{\|e_1\|}{T}\cos(\alpha_1 - \omega_1 T).$$

Similarly, for the feedforward terms in the angular velocity of controllers (6) and (7), we have

$$\dot\theta_e \approx \omega_1 + \frac{\alpha_2 - \alpha_1}{T}$$

while $\dot\theta_b$ for controller (8) can be retrieved using two subsequent estimates of the alignment error. Both these estimates are computed at t_2.

The ball localization procedure shown so far proceeds from the surmise that the ball is at the center of the image plane. While the majority of simulation and experimental results to be presented in Section 5 will support the adoption of this simplifying assumption, the use of the associated ball localization procedure may be ineffective if the camera tracking performance deteriorates (e.g., in the presence of very fast target dynamics), and may even prevent the successful completion of the interception task.

However, the generalization of the ball localization procedure to the case of nonperfect camera tracking is readily obtained. As shown by Fig. 5, γ and η (given by (3) and (5)) should be used to identify the ball position whenever the ball offset ξ is nonzero. Hence, it is sufficient to use these angles in place of φ and ψ in (9)–(11), thereby obtaining the new expressions

$$b = \frac{h}{\tan\eta} \tag{12}$$

$$\|e\|^2 = b^2 + s^2 - 2bs\cos(\pi - \gamma) \tag{13}$$

$$\alpha = \arcsin\frac{b\sin(\pi - \gamma)}{\|e\|}. \tag{14}$$

All the remaining computations are unchanged.

Note that all the previous computations only use the pan–tilt angles and the ball offset in the image plane; as a consequence, our control scheme is image-based and does not need odometric information on the robot location. More sophisticated versions of the above estimators can be designed, also taking into account target dynamics and/or filtering the estimated quantities to account for the presence of noise.

5. Simulations and experiments

Before proceeding to an experimental validation of our method, a simulation study was carried out to assess its performance. First, MATLAB was used for assessing the interception performance of controllers (6)–(8) in themselves. Second, the complete camera-robot model was simulated in Webots to verify that a visual implementation of the interception schemes was indeed possible. Finally, experiments on a MagellanPro robot were performed.

5.1. Matlab simulations

A comparison of the various controllers discussed in Section 4.1 was made within MATLAB, assuming that all the relevant variables and their derivatives were directly available. Hence, the camera tracking subsystem is not simulated at this stage. Typical results are shown in Figs. 9 and 11, where the robot and the ball are represented as a triangle and a circle, respectively. Plots (a)–(c) show the results obtained by applying control laws (6)–(8), respectively. As suggested in [3], the following gains were used for the control law (8):

$$k_1 = k_2 = 2\zeta\sqrt{\dot\theta_b^2 + \beta v_b^2} \quad k_3 = \beta \tag{15}$$

where $\beta = 1$ and $\zeta = 0.7$. In order to compare the different controller performances, the above gain values k_1 and k_2 were also used for controllers (6) and (7). As for the third control gain, we set $k_3 = \beta|v_b|$ for consistency, since in this case the corresponding feedback term does not contain the ball velocity norm.

In Fig. 9 the ball is moving along a line with a constant speed of 1 m/s. The absence of the feedback term $e^{\mathrm{T}} n_\perp$ in controller (6) (plot (a)) results in a slightly less efficient interception with respect to controller (7) (plot (b)), which drives the robot more effectively to the ball. This difference can be emphasized by increasing k_3.

On the other hand, controller (8) (plot (c)) shows a marked tracking (as opposed to interception) attitude, consistent with the use of the alignment error $\theta_b - \theta$ in place of the pointing error $\theta_e - \theta$. Note the small motion inversion midway along the path, due to the projection of e along the robot forward axis becoming negative and larger than the projection of v_b.

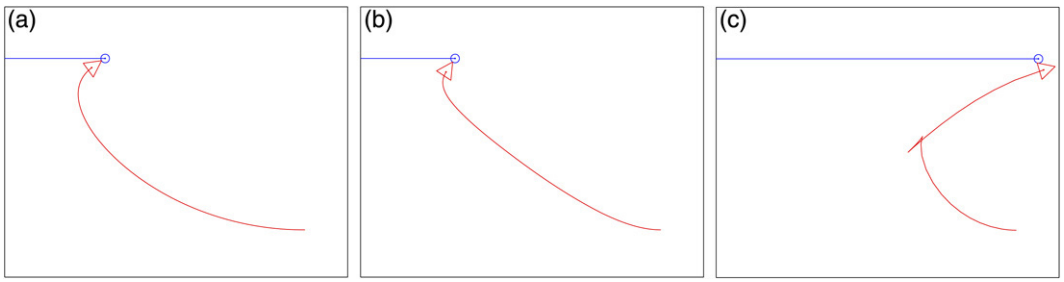

Fig. 9. First MATLAB simulation: robot trajectories as the ball moves along a linear path: (a) controller (6); (b) controller (7); (c) controller (8).

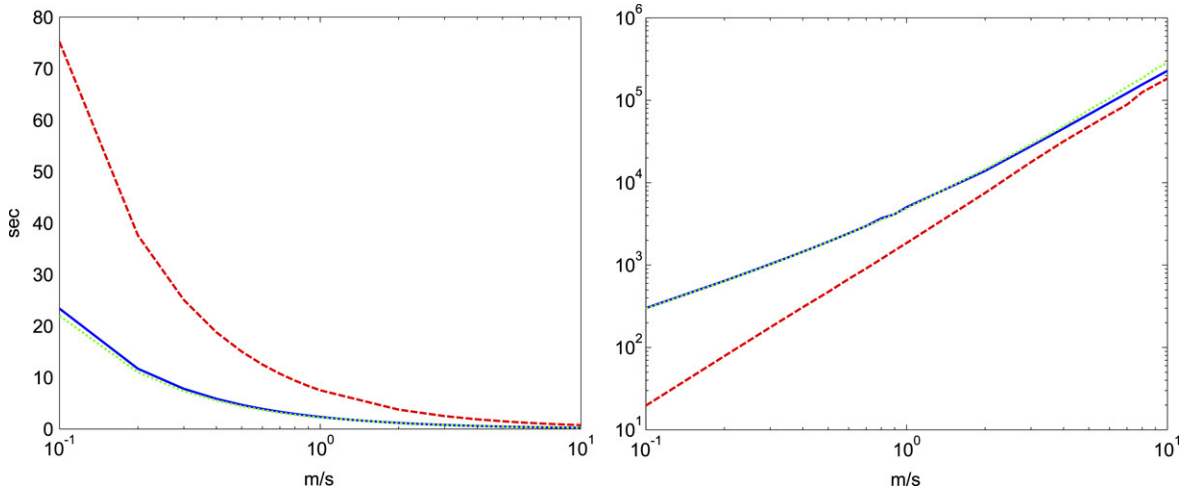

Fig. 10. Results obtained for a ball moving along a linear path with velocity varying between 0.1 and 10 m/s. *Left*: Time needed for interception. *Right*: Control effort. Continuous line (blue): controller (6). Dotted line (green): controller (7). Dashed line (red): controller (8).

Fig. 10 summarizes the results obtained with controllers (6)–(8) for a ball moving along a linear path with constant velocity varying between 0.1 and 10 m/s, in terms of the time needed for interception and the control effort (integral of the squared velocity norm). The control gains were chosen as previously described. Controllers (6) and, in particular, (7) were much faster in intercepting slow balls, whereas (8) was found to be slightly less demanding in terms of control effort, at least for this choice of control gains. These results can be explained by considering that, for low ball velocities, the tracking attitude of controller (8) delays the interception, privileging instead the alignment between the robot and target path. For high ball velocities, the performance tends to become the same, because the ball gets quickly ahead of the robot and all three controllers have no choice but to follow the trail.

Similar results are obtained in the second simulation, shown in Fig. 11, where the target moves along a sinusoidal path with a constant speed of 2 m/s. Controllers (6) and (7), designed for interception, constantly push the robot toward the target, trying to reduce the relative distance. On the other hand, controller (8), designed for tracking, tends to align the robot with the ball velocity, resulting in a robot trajectory which *coasts* the target. Again, with this controller, a motion inversion occurs during the approach.

A final simulation was performed to highlight the different behavior of the controllers when the target stops along its path; see Fig. 12. In this case, all control gains were set to a constant unit value (note that, with the previous choice (15), k_1 and k_2 would go to zero when the ball stops). While controller (7) keeps moving the robot and leads to a successful interception, under the action of controller (8) the robot performs parallel parking w.r.t. the ball direction and then stops. This is due to the fact that when the target velocities go to zero, Eq. (8) gives $v = k_1 e^{\mathrm{T}} n_\theta$ and $\omega = k_2(\theta_b - \theta)$, making it impossible to recover an orthogonal error vector.

Note that all the presented simulations were interrupted at the time of interception, assuming that the robot and the ball have a nonzero size. However, by letting the simulation run indefinitely, one may verify that the alignment error $\theta_b - \theta$ asymptotically converge to zero also under controllers (6) and (7), provided that the target does not stop. On the other hand, the pointing error $\theta_e - \theta$ does not tend to zero with controller (8). Roughly speaking, we could say that (6) and (7) are primarily interception, but also tracking, controllers, while (8) provides pure tracking.

5.2. *Webots simulations*

To perform a more realistic simulation, accounting also for the presence of the pan–tilt camera system, our visual control scheme was implemented in Webots.[3] The simulated camera,

[3] Webots is a commercial mobile robot simulation software developed by Cyberbotics Ltd (http://www.cyberbotics.com).

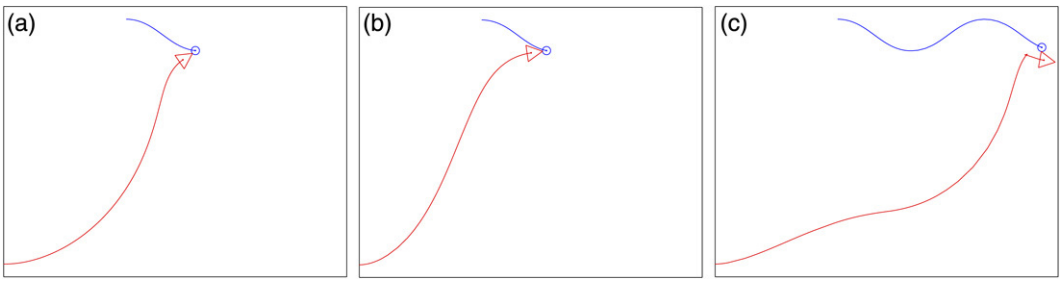

Fig. 11. Second MATLAB simulation: robot trajectories as the ball moves along a sinusoidal path: (a) controller (6); (b) controller (7); (c) controller (8).

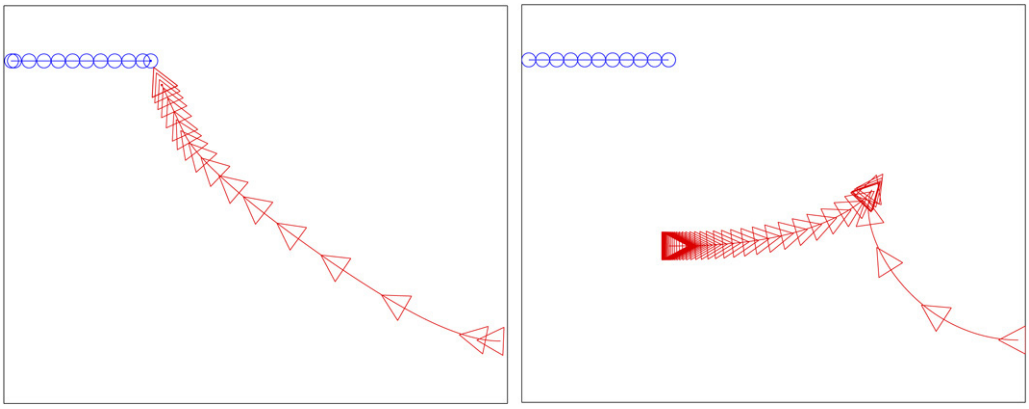

Fig. 12. Third MATLAB simulation: robot trajectories for a ball which suddenly stops. *Left*: controller (7). *Right*: controller (8).

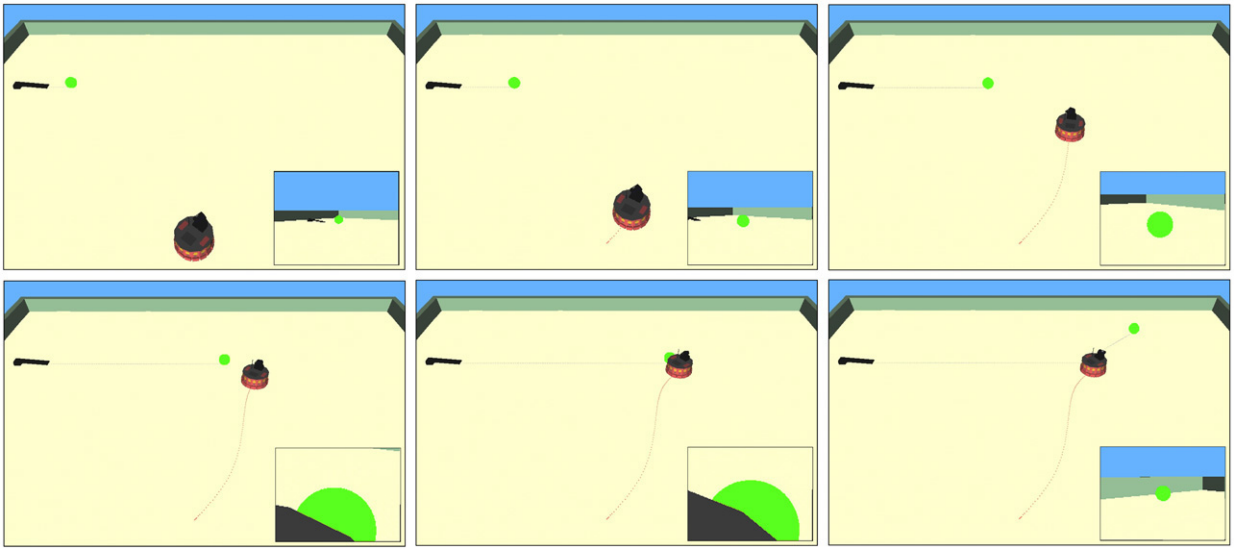

Fig. 13. First Webots simulation: interception of a ball moving along a linear path using the camera tracking commands (2) and (4) in conjunction with controller (7). Ball localization is performed under the simplifying assumption of perfect camera tracking.

which reproduces exactly the characteristics (including noise) of the EVI-D31 mounted on MagellanPro, acquires scene images which are then processed using the techniques presented in Section 3 in order to recognize the ball and keep it close to the center of the image. At this preliminary stage, the simulated acquisition frame-rate has been set to a higher value (31 Hz) than the actual one (7 Hz) in order to have the same sampling time for both the robot and camera control systems. These settings, with the application of the camera control laws (2) and (4), yield an effective camera tracking scheme whose dynamics is much faster than the robot's. The various control laws are then visually implemented, as explained in Section 4.2, with the achievable robot linear and angular velocities bounded respectively at 2.5 m/s and 3 rad/s.

A first simulation result obtained with controller (7) is shown in Fig. 13. The ball moves along a linear path at a constant speed of 2 m/s. Ball localization is performed under the simplifying assumption of perfect camera tracking, i.e., using (9)–(11). As a matter of fact, the camera views, visible in the bottom-right corner of each snapshot, show that the ball is not perfectly

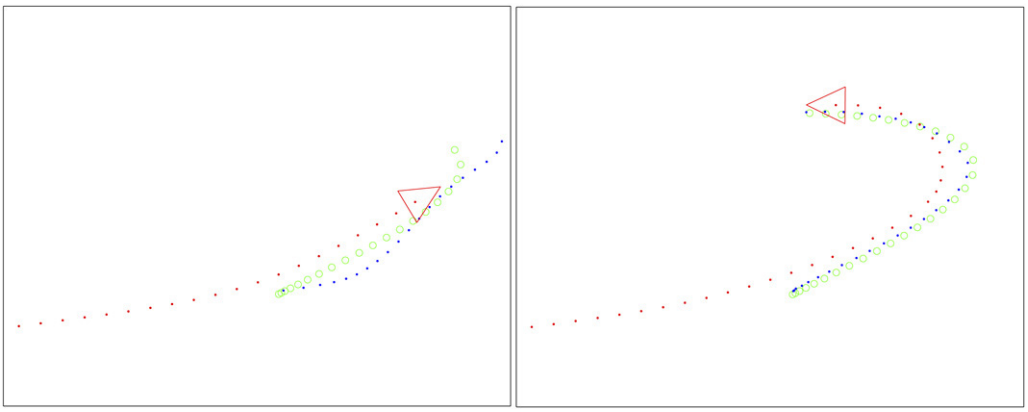

Fig. 14. Second Webots simulation: interception of a ball moving along a curved path using the camera tracking commands (2) and (4) with no feedforward, in conjunction with controller (7). Circles (green) and dots (blue) represent the actual and estimated ball trajectory, respectively, while the red triangle represents the final robot configuration. *Left*: unsuccessful interception when the ball is localized using the simplifying assumption of perfect camera tracking. *Right*: successful interception when the ball is accurately localized.

centered with respect to the image, and even partially occluded by the robot own body at close range. Nonetheless, interception is successfully completed, confirming that the proposed two-level scheme is quite robust with respect to nonidealities. Similar results are obtained for the other controllers (6) and (8), whose fundamental behavior illustrated in Section 4.1 is preserved.

In this case, we have verified that the interception trajectories obtained using (12)–(14) for accurate ball localization are not significantly different from the above trajectories. This is essentially due to the fact that the ball is traveling along a line at relatively low speed.

To highlight the possible benefit coming from the use of accurate ball localization, we have run in Webots another simulation which is more challenging with respect to camera tracking. To this end, we introduced the following modifications: the ball moves along an eight-shaped path at a constant speed of 2 m/s, and the feedforward terms in the camera tracking commands (2) and (4) were omitted. Results are shown in Fig. 14. On the left, localization is performed via (9)–(11): as a result, camera tracking is ineffective in correspondence of the maximum curvature of the path, where the ball leaves the field of view, and interception does not occur. On the right, successful interception is obtained by the same controller when accurate ball localization via (12)–(14) is used.

5.3. Experiments

We now present some experimental results obtained with the MagellanPro robot available in our lab using the interception controller (7) with the camera tracking commands (2) and (4). Sampling time was set to 0.13 s, i.e., exactly the refresh time of the camera frame grabber (see Section 2). Fig. 15 illustrates a successful experiment by means of superimposed snapshots. In this case, the ball was traveling along an approximately linear path with a speed of 0.4 m/s, and the controller gain values are $k_1 = 0.5$, $k_2 = 1.2$, $k_3 = 0.2$, $k_\varphi = 0.2$ and $k_\psi = 0.1$. During this kind of test, a sliding ramp was used in order to guarantee repeatable ball initial conditions.

A more challenging experiment is shown in Fig. 16. Here, the ball was passed back and forth by two human players in a manner reminiscent of a common soccer training exercise. The robot does a good job in hunting and finally intercepting the ball, in spite of its frequent motion reversals. Movie clips of these experiments, as well as of Webots simulations, are available at http://www.dis.uniroma1.it/~labrob/research/VisualInterception.html.

In both experiments, accurate ball localization was not needed. This is essentially due to the fact that the ball is traveling along linear paths at relatively low speeds.

A comment is in order concerning the practical possibility that the ball leaves the camera field of view due, e.g., to the low acquisition frame rate or the insufficient speed of the camera tracking dynamics w.r.t. sudden changes in the ball trajectory (for example, this happens in the second experiment shown above). Through repeated experiments, it has been verified that an effective solution is to keep the last camera pan–tilt commands (i.e., those computed during the last successful ball detection) until the image of the ball is recovered, or until a certain number of sequential frames are acquired without seeing the ball. In this second case, the robot stops and scans the environment with the pan–tilt camera looking for the ball.

6. Conclusions and future work

We have presented a two-level vision-based scheme for driving a nonholonomic mobile robot to intercept a moving target. On the lower level, the pan–tilt platform carrying the on-board camera is controlled so as to keep the target as close as possible to the center of the image plane. On the higher level, the robot computes the relative position of the target on the basis of its image coordinates and the pan–tilt angles, and moves under the action of a control law aimed at interception. Various possible choices have been discussed and rigorously analysed for the high-level robot controller. The proposed visual interception method has been validated through simulations as well as experiments on the mobile robot

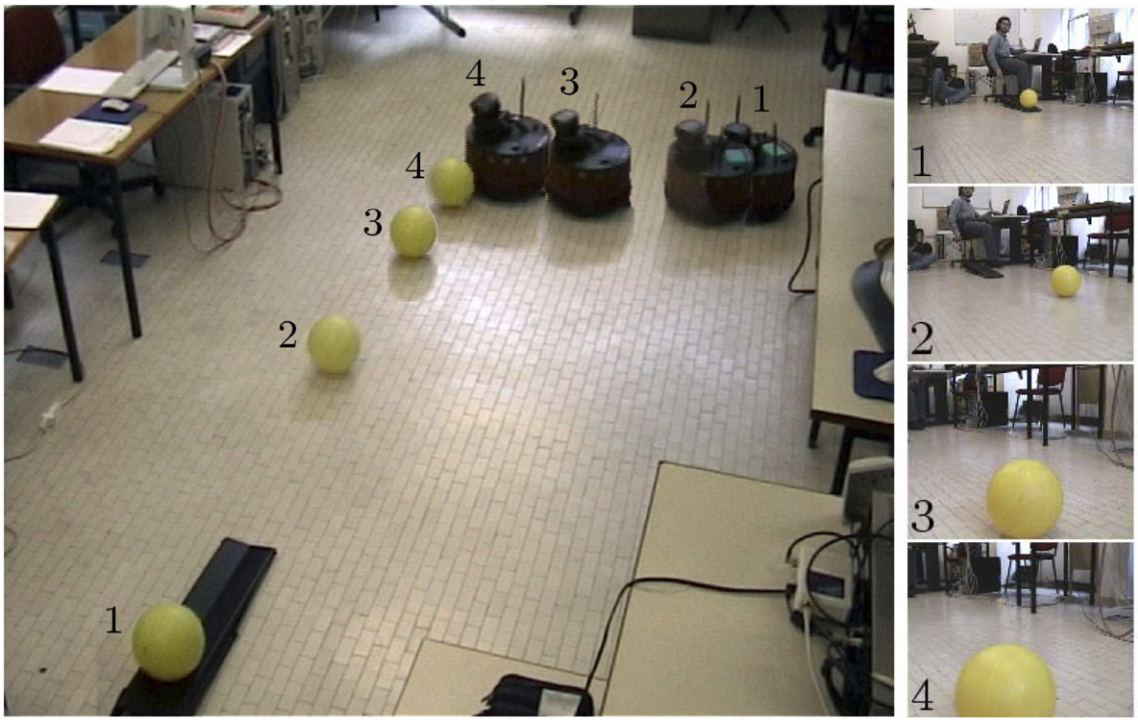

Fig. 15. A successful interception experiment with the ball traveling along a linear path. The numbers identify the position of the ball and the robot at four consecutive time instants.

MagellanPro, confirming that the method is quite robust to actual system limitations and nonidealities.

We are currently pursuing several research directions to extend the proposed control scheme, including the addition of a Kalman filter to provide robust estimates of the ball position and velocity, and an extension of the method in order to allow the interception along a specified direction, motivated by the robot soccer application.

We conclude by relating the results of this paper to the well-known problem of the nonexistence of *universal* continuous feedback stabilizers (i.e., controllers which guarantee exact tracking of arbitrary trajectories, either persistent or nonpersistent) for nonholonomic systems [13]. Controllers (6) and (7) are not exceptions, for their tracking control capability is lost when the target stops (see the Appendix, Remark 3); still, interception will be accomplished. Another way to obtain a similar result would be to resort to smooth feedback controllers which achieve *practical* stabilization, i.e., ultimately bounded tracking error, for arbitrary trajectories [19].

Appendix

Here we provide proofs of the asymptotic stability of the error dynamics with controllers (6) and (7). To this end, we assume that the target moves as a unicycle:

$$\begin{aligned}\dot{x}_b &= v_b \cos\theta_b \\ \dot{y}_b &= v_b \sin\theta_b \\ \dot{\theta}_b &= \omega_b.\end{aligned} \tag{16}$$

Note that a unicycle can perform quite general planar trajectories, i.e., those that can be described by two parametric functions $x_b(t)$, $y_b(t)$ of class $\bar{C}^2$. Also, given one such trajectory, it is possible to compute algebraically the corresponding open-loop commands v_b and ω_b (a property known as *flatness*).

Define the following state error:

$$\varepsilon = \begin{pmatrix}\varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3\end{pmatrix} = \begin{pmatrix}\cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1\end{pmatrix} \begin{pmatrix}x_b - x \\ y_b - y \\ \theta_e - \theta\end{pmatrix}. \tag{17}$$

Note that ε_1 and ε_2 represent the components of the cartesian error vector e in a rotated frame which is aligned with the robot forward axis, while ε_3 is the pointing error (refer to Fig. 2). Hence, we have:

$$\|e\|^2 = \varepsilon_1^2 + \varepsilon_2^2, \quad \varepsilon_1 = \|e\| \cos\varepsilon_3, \quad \varepsilon_2 = \|e\| \sin\varepsilon_3.$$

Using (17), (1) and (16), the dynamics of ε can be expressed as

$$\begin{aligned}\dot{\varepsilon}_1 &= \omega\varepsilon_2 - v + v_b \cos(\theta_b - \theta) \\ \dot{\varepsilon}_2 &= -\omega\varepsilon_1 + v_b \sin(\theta_b - \theta) \\ \dot{\varepsilon}_3 &= \dot{\theta}_e - \omega.\end{aligned} \tag{18}$$

Proposition 1. *Assume that v_b is bounded, i.e., $|v_b(t)| < v_M$, $\forall t$. Then, controller (6) globally asymptotically stabilizes the origin $\varepsilon = 0$.*

Proof. First, it will be shown that $\varepsilon \to 0$ as $t \to \infty$; then, the Lyapunov stability of the system will be proved.

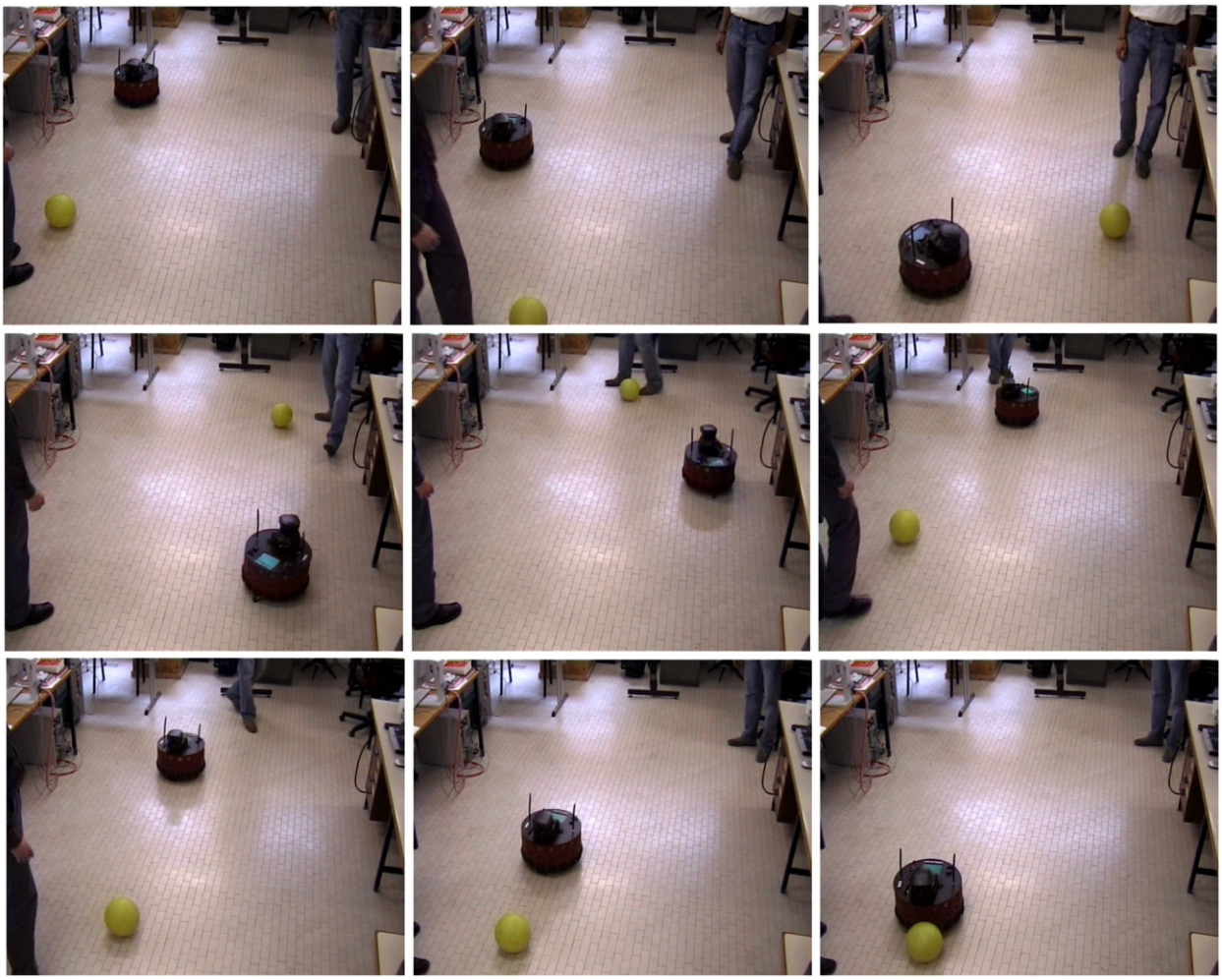

Fig. 16. An experiment where the robot tracks and intercepts a ball passed back and forth by two human players.

Under the control law (6) the dynamics becomes:

$$\begin{aligned}\dot{\varepsilon}_1 &= k_2\varepsilon_2\varepsilon_3 + \dot{\theta}_e\varepsilon_2 - k_1\varepsilon_1\\ \dot{\varepsilon}_2 &= -k_2\varepsilon_1\varepsilon_3 - \dot{\theta}_e\varepsilon_1 + v_b\sin(\theta_b-\theta)\\ \dot{\varepsilon}_3 &= -k_2\varepsilon_3.\end{aligned} \tag{19}$$

Clearly, ε_3 converges to zero exponentially. Let

$$V_1 = \frac{1}{2}(\varepsilon_1^2+\varepsilon_2^2) = \frac{1}{2}\|e\|^2 \tag{20}$$

which has the following time derivative along the trajectory of (19):

$$\begin{aligned}\dot{V}_1 &= -k_1\|e\|^2\cos^2\varepsilon_3 + \|e\|v_b\sin\varepsilon_3\sin(\theta_b-\theta)\\ &\le F(e,\varepsilon_3)\end{aligned} \tag{21}$$

where we have set

$$F(e,\varepsilon_3) = -k_1\|e\|^2\cos^2\varepsilon_3 + \|e\||\sin\varepsilon_3|v_M.$$

Then, defining

$$R(\varepsilon_3) = \frac{|\sin\varepsilon_3|v_M}{k_1\cos^2\varepsilon_3}$$

we have $\dot{V}_1 \le F(e,\varepsilon_3) < 0$ whenever $\|e\| > R(\varepsilon_3)$. Note that $R(\varepsilon_3)\to 0$ as $t\to\infty$.

Choose an arbitrary $a > 0$. Call $\bar{t}$ the first instant such that $R(\varepsilon_3) \le a$ (if $R(\varepsilon_3(t_0)) \le a$, then $\bar{t} = t_0$; if $R(\varepsilon_3(t_0)) > a$, then $\bar{t} > t_0$ and $R(\varepsilon_3(\bar{t})) = a$). We have two cases:

1. $\|e(\bar{t})\| > a$. Since in this case $\|e(t)\| > a \ge R(\varepsilon_3)$, $\dot{V}_1$ is negative and bounded above until $\|e\| = a$ at some time t_a; for $t > t_a$, $\|e\|$ cannot exceed a, because $\dot{V}_1$ is negative outside the ball of radius a (as well as on its boundary).
2. $\|e(\bar{t})\| \le a$. In this case, $\dot{V}_1(\bar{t})$ may be non-negative, but $\|e\|$ cannot exceed a for $t > \bar{t}$ because $\dot{V}$ is negative outside the ball of radius a and zero on its boundary.

Hence, $\forall a$, there exists a time t_a ($=\bar{t}$ in case 2) such that $\|e(t)\| \le a$ for $t \ge t_a$; i.e., $\|e\|$ converges to zero.

It remains to be shown that the origin is Lyapunov stable, i.e.,

$$\forall\eta > 0, \quad \exists\delta_\eta > 0 : \|\varepsilon(t_0)\| < \delta_\eta \Rightarrow \|\varepsilon(t)\| < \eta, \quad \forall t > t_0.$$

Choose $\varepsilon(t_0)$ such that $\|e(t_0)\| < \eta/2$, $|\varepsilon_3(t_0)| < \eta/2$ and $R(\varepsilon_3(t_0)) < \eta/2$ (this can always be done because $R(\varepsilon_3)\to 0$ as $\varepsilon_3 \to 0$). Then, $\|e\|$ cannot exceed $\eta/2$ because $\dot{V} < 0$ outside the ball of radius $\eta/2$ (as well as on its boundary). On

the other hand, ε_3 is exponentially converging to zero, so that $\varepsilon_3(t)$ never exceeds $\eta/2$. Hence, $\|\varepsilon\|$ never exceeds η. □

Proposition 2. *Assume that v_b is bounded, i.e., $|v_b(t)| < v_M$, $\forall t$. Then, controller (7) globally asymptotically stabilizes the origin $\varepsilon = 0$.*

Proof. Under the controller (7), the dynamics becomes:

$$\begin{aligned}\dot{\varepsilon}_1 &= k_2\varepsilon_2\varepsilon_3 + k_3\varepsilon_2^2 + \dot{\theta}_e\varepsilon_2 - k_1\varepsilon_1\\ \dot{\varepsilon}_2 &= -k_2\varepsilon_1\varepsilon_3 - k_3\varepsilon_1\varepsilon_2 - \dot{\theta}_e\varepsilon_1 + v_b\sin(\theta_b-\theta)\\ \dot{\varepsilon}_3 &= -k_2\varepsilon_3 - k_3\varepsilon_2.\end{aligned} \tag{22}$$

Also in this case, ε_3 exponentially tends to zero. In fact, consider the function

$$V_2 = \frac{1}{2}\varepsilon_3^2$$

which has the following time derivative along the trajectories of (22):

$$\dot{V}_2 = -k_2\varepsilon_3^2 - k_3\|e\|\varepsilon_3\sin\varepsilon_3.$$

Since $\varepsilon_3\sin\varepsilon_3 \geq 0$, $\forall\varepsilon_3$, it follows that $\dot{V}_2 \leq -k_2\varepsilon_3^2 < 0$ and hence the exponential convergence of ε_3.

Now consider again the function V_1 defined by (20), and note that under (22) it has the same time derivative (21). Hence, the same arguments of the previous proof can be used to show asymptotic stability. □

Some additional remarks are in order with respect to the above analysis.

Remark 1. It is possible to show that the convergence of ε towards zero is always faster with controller (7) than with controller (6). In fact, define the following positive definite function:

$$V_3 = \frac{1}{2}\|\varepsilon\|^2.$$

Using Eq. (18), the time derivative of V_3 under controller (6) is

$$\begin{aligned}\dot{V}_{3(6)} &= -k_1\|e\|^2\cos^2\varepsilon_3 + \|e\|v_b\sin\varepsilon_3\sin(\theta_b-\theta) - k_2\varepsilon_3^2\\ &\leq F(e,\varepsilon_3) - k_2\varepsilon_3^2\end{aligned}$$

while under controller (7) we have

$$\begin{aligned}\dot{V}_{3(7)} &= -k_1\|e\|^2\cos^2\varepsilon_3 + \|e\|v_b\sin\varepsilon_3\sin(\theta_b-\theta) - k_2\varepsilon_3^2\\ &\quad - k_3\|e\|\varepsilon_3\sin\varepsilon_3 \leq F(e,\varepsilon_3) - k_2\varepsilon_3^2 - k_3\|e\|\varepsilon_3\sin\varepsilon_3.\end{aligned}$$

Since $\dot{V}_{3(7)} \leq \dot{V}_{3(6)}$, $\forall\varepsilon$, we conclude that the convergence of ε to zero is faster with controller (7) than with controller (6).

Remark 2. Note that, differently from the proof of asymptotic stability of controller (8) in [22], persistency of the target trajectory was not assumed in Propositions 1 and 2. In fact, this is not needed: if the target stops, convergence of ϵ to zero is still guaranteed with controllers (6) and (7) (see the third simulation in Section 5.1).

Remark 3. It is possible to show analitically that, if the target trajectory does not stop, controllers (6) and (7) also drive the alignment error $\theta_b - \theta$ to zero. The same result can also be intuitively inferred from the unicycle kinematics, for it is clear that in order to match asymptotically the position ($e \to 0$) of a *moving* unicycle it is necessary to match its orientation as well.

References

[1] G. Artus, P. Morin, C. Samson, Tracking of an omnidirectional target with a nonholonomic mobile robot, in: 2003 Int. Conf. on Advanced Robotics, 2003, pp. 1468–1473.

[2] J.A. Borgstadt, N.J. Ferrier, Interception of a projectile using a human vision-based strategy, in: 2000 IEEE Int. Conf. on Robotics and Automation, 2000, pp. 3189–3196.

[3] C. Canudas de Wit, H. Khennouf, C. Samson, O.J. Sørdalen, Nonlinear control design for mobile robots, in: Y.F. Zheng (Ed.), Recent Trends in Mobile Robots, vol. 11, World Scientific Publisher, 1993, pp. 121–156.

[4] P.I. Corke, Visual Control of Robots: High Performance Visual Servoing, RSP Press, 1996.

[5] C. Coué, P. Bessière, Chasing an elusive target with a mobile robot, in: 2001 IEEE Int. Conf. on Robotics and Automation, 2001, pp. 1370–1375.

[6] P.I. Corke, D. Symeonidis, K. Usher, Tracking road edges in the panospheric image plane, in: 2003 IEEE/SRJ Int. Conf. on Intelligent Robots and Systems, 2003, pp. 1330–1335.

[7] A.K. Das, R. Fierro, V. Kumar, B. Southall, J. Spletzer, C.J. Taylor, Real-time vision-based control of a nonholonomic mobile robot, in: 2001 IEEE Int. Conf. on Robotics and Automation, 2001, pp. 1714–1719.

[8] A. De Luca, G. Oriolo, Local incremental planning for nonholonomic mobile robots, in: 1994 IEEE Int. Conf. on Robotics and Automation, 1994, pp. 104–110.

[9] M.K. Hu, Visual pattern recognition by moment invariants, IRE Transactions on Information Theory 8 (1962) 179–187.

[10] D. Hujic, E.A. Croft, G. Zak, R. Fenton, J.K. Mills, B. Benhabib, The robotic interception of moving objects in industrial settings: Strategy development and experiment, IEEE Transactions on Mechatronics 3 (1998) 225–239.

[11] H. Lausen, J. Nielsen, M. Nielsen, P. Lima, Model and behavior-based robotic goalkeeper, in: RoboCup 2003: Robot Soccer World Cup VII, 2003, pp. 169–180.

[12] R.N. Lea, Y. Jani, Design and performance comparison of fuzzy logic based tracking controllers, in: 1st Int. Joint Conf. on Fuzzy Logic (NAFIPS/IFIS/NASA'94), 1994, pp. 340–344.

[13] D.A. Lizárraga, Obstructions to the existence of universal stabilizers for smooth control systems, Mathematics of Control, Signals and Systems 16 (4) (2004) 255–277.

[14] Y. Masutani, M. Mikawa, N. Maru, F. Miyazaki, Visual servoing for non-holonomic mobile robots, in: 1994 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 1994, pp. 1133–1140.

[15] G.L. Mariottini, G. Oriolo, D. Prattichizzo, Image-based visual servoing for nonholonomic mobile robots using epipolar geometry, IEEE Transactions on Robotics and Automation 23 (2007) 87–100.

[16] C.H. Messom, G.S. Gupta, S. Demidenko, L.Y. Siong, Improving predictive control of a mobile robot: Application of image processing and kalman filtering, in: 2003 Instrumentation and Measurement Technology Conf., 2003, pp. 1492–1496.

[17] M. Mehrandezh, N.M. Sela, R.G. Fenton, B. Benhabib, Robotic interception of moving objects using an augmented ideal proportional navigation guidance technique, IEEE Transactions on Systems, Man and Cybernetics, Part A 30 (2000) 238–250.

[18] R. Mori, F. Miyazaki, GAG (Gaining Angle of Gaze) strategy for ball tracking and catching task, in: 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2002, pp. 281–286.

[19] P. Morin, C. Samson, Practical stabilization of driftless systems on Lie Groups: The transverse function approach, IEEE Transactions on Automatic Control 48 (2003) 1496–1508.
[20] K. Mu, T.G. Sugar, M.K. McBeath, Perceptual navigation strategy: A unified approach to interception of ground balls and fly balls, in: 2003 IEEE Int. Conf. on Robotics and Automation, 2003, pp. 3461–3466.
[21] J.C. Russ, The Image Processing Handbook, CRC Press, 2002.
[22] C. Samson, K. Ait-Abderrahim, Feedback control of a nonholonomic wheeled cart in cartesian space, in: 1991 IEEE Int. Conf. on Robotics and Automation, 1991, pp. 1136–1141.
[23] D.P. Tsakiris, P. Rives, C. Samson, Applying visual servoing techniques to control nonholonomic mobile robots, in: 1997 Int. Conf. on Intelligent Robots and Systems (Workshop on New Trends in Image-Based Robot Servoing), 1997.
[24] C.-D. Yang, F.-B. Hsiao, F.-B. Yeh, Generalized guidance law for homing missiles, IEEE Transactions on Aerospace and Electronic Systems 25 (1989) 197–212.
[25] H. Zhang, J.P. Ostrowski, Visual motion planning for mobile robots, IEEE Transactions on Robotics and Automation (2002) 199–208.

Luigi Freda received the M.Sc. degree in computer engineering from the University of Rome "La Sapienza", Italy, in 2003. He is currently a Ph.D. student (expected graduation in 2007) at the same university. His research interests lie in the areas of robot exploration, motion planning, minimal sensing and control of mobile robots.

Giuseppe Oriolo received the Ph.D. degree in Control Systems Engineering in 1992 from the University of Rome "La Sapienza". Since 1998, he is an *Associate Professor* of Automatic Control at the same university, where he is in charge of the Robotics Laboratory. His research interests are in the area of robotics and control, in which he has published more than 100 papers. He is a Senior Member of IEEE.