CrossMark

Vision-Based Maze Navigation for Humanoid Robots

A. Paolillo · A. Faragasso · G. Oriolo · M. Vendittelli

Abstract We present a vision-based approach for navigation of humanoid robots in networks of corridors connected through curves and junctions. The objective of the humanoid is to follow the corridors, walking as close as possible to their center to maximize motion safety, and to turn at curves and junctions. Our control algorithm is inspired by a technique originally designed for unicycle robots that we have adapted to humanoid navigation and extended to cope with the presence of turns and junctions. In addition, we prove here that the corridor following control law provides asymptotic convergence of robot heading and position to the corridor bisector even when the corridor walls are not parallel. A State Transition System is designed to allow navigation in mazes of corridors, curves and T-junctions. Extensive experimental validation proves the validity and robustness of the approach.

Keywords vision-based navigation · humanoid robots · visual control

1 Introduction

In the last decade humanoid robots have increasingly attracted the interest of researchers for the challenging

G. Oriolo · A. Paolillo · M. Vendittelli
Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Roma, Italy.
Tel.: +39-06-77274049
Fax: +39-06-77274033
E-mail: {oriolo, paolillo, vendittelli}@diag.uniroma1.it

A. Faragasso
Centre for Robotics Research, Department of Informatics, Kings College London, London, WC2R 2LS,UK
E-mail: {angela.faragasso}@kcl.ac.uk

scientific problems that they present and for their impact in real applications. The high flexibility and dexterity of these systems make them particularly suitable for real word tasks hardly executable by other kind of robots. After an initial period during which research has been focused on fundamental issues related, e.g., to stability of locomotion, composition of motion and manipulation tasks, whole body motion, motion planning and other basic problems, a big attention is now being devoted to the exploitation of the exteroceptive sensing capabilities of these complex robotic systems with the aim of reaching their full autonomy. The current availability of experimental platforms with reasonable cost and built-in basic functionalities, like robust and reactive locomotion systems, allows to investigate and design sensor-based control paradigms that are sufficiently abstract to be portable on different humanoid robots.

In this paper we address the problem of humanoid navigation in indoor environments consisting in networks of corridors using visual information. Related literature proposes vision-based methods enabling indoor navigation of humanoids based on odometric (Oriolo et al, 2013) or map-based localization (Lutz et al, 2012; Back et al, 2012; Delgado-Galvan et al, 2015), relying on preregistered images of the environment (Ido et al, 2009; Delfin et al, 2014) or using known landmarks (George and Mazel, 2013; Wei et al, 2014). Finally, other methods achieve indoor navigation among obstacles by using planning techniques (Maier et al, 2013, 2012; Kuffner Jr et al, 2001; Sabe et al, 2004).

Alternative strategies need to be investigated when no localization system nor a priori information about the environment is available. In this paper we propose an Image-Based Visual Servoing (IBVS) approach to generate walking commands that force the humanoid

to walk at the center of corridors using only visual information, and natural features, and with no a priori knowledge of the environment.

Previous work based on the visual servoing paradigm for generating walking motion include Dune et al (2011), where a Position-Based Visual Servoing (PBVS) has been used to drive the humanoid to specific locations, the work of Courty et al (2001) that proposes an Image-Based Visual Servoing (IBVS) algorithm for the animation of human characters, or (Garcia et al, 2014), where visual information is mapped to a walking pattern generator to achieve image features regulation through a PBVS or an IBVS control scheme.

The goal of the present paper is to achieve navigation tasks in indoor and office-like environments represented as a network of corridors through which the humanoid should move without colliding with the corridors walls.

In a previous work (Faragasso et al, 2013) we have used a IBVS technique to attain this goal. The method was based on (Toibero et al, 2009), a corridor following method with proved heading and position convergence which does not require a map of the environment nor the localization of the robot inside the corridor. In (Faragasso et al, 2013) we have adapted this algorithm to the navigation of humanoid robots and generalized the control law to allow handling of curves and junctions. The resulting method avoids switching among different controllers (Vassallo et al, 2000; Cherubini et al, 2008) or complex image processing (Park and Suh, 2010; Matsumoto et al, 2000; Ohnishi and Imiya, 2013) to achieve navigation tasks like turning around a corner. Furthermore, the exclusive use of monocular vision, prevents algorithmic complexity induced by multi-sensor architectures (Tello Gamarra et al, 2005).

In the present paper, beside deepening the analysis of the results presented in (Faragasso et al, 2013), we formally prove that the visual-based control law therein proposed for the navigation of humanoid robots in corridors with parallel walls is actually robust with respect to walls relative slopes. To show this, we derive the dynamics of the visual features in the general case of non-parallel corridor walls and prove that closed loop convergence of robot heading and position is preserved.

To illustrate the effectiveness of the method for navigation in complex, maze-like environments we propose the design of a State Transition System (STS) representing networks of corridors with arbitrary relative inclination of the wall guidelines and connected through curves and T-junctions. The results are supported by experimental validation.

Paper organization is as follows. In Sect. 2 we formulate the problem and briefly illustrate the proposed
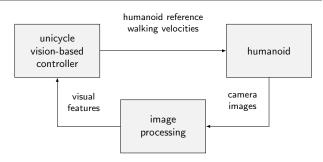

Fig. 1: Conceptual illustration of the proposed control approach for corridor navigation of humanoid robots.

approach. Section 3 presents the visual features mathematical model to be used in the derivation of the control law which is presented in Sect. 4, while Sect. 5 illustrates the strategy adopted to allow navigation in maze-like environments. In Sect. 6 we explain how the formal results reported in the previous sections, and making reference to an ideal mobility model of the humanoid corresponding to that of a unicycle, extend to humanoid robots. The experimental validation results are presented in Sect. 7 and in the accompanying video. Section 8 offers concluding comments and a perspective on future work. Finally, the appendix provides the robustness proof for the proposed controller.

2 Control objective and approach

Consider a humanoid whose task requires navigation on flat grounds and in a maze-like environment, i.e., a network of rectilinear corridors connected through corners and junctions. Our objective is to design a visual-based controller enabling the robot to walk as close as possible to the corridors center and to negotiate connections for proceeding toward a desired direction possibly specified by a higher-level module.

The proposed approach relies on a visual servoing scheme developed for a unicycle which is adopted as the humanoid mobility model. Experimental studies (Mombaur et al, 2010; Truong et al, 2010) on human locomotion have, in fact, shown that, on long distance walks in uncluttered environments, the orientation of human sagittal plane is tangent to the path for most of the time. In other words, this kind of human paths resemble very closely those typical of nonholonomic wheeled mobile robots such as the unicycle.

The problem considered in this paper fits exactly the long distance walk scenario. Motivated by this, and assuming that the walking speed and direction of the humanoid can be controlled by sending reference velocities to its locomotion system, we use, as mobility
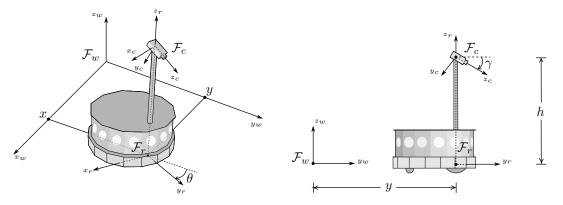
Fig. 2: Frames of interest for derivation of the visual features model and design of the vision-based controller: world ($\mathcal{F}_w$), robot ($\mathcal{F}_r$) and camera frame ($\mathcal{F}_c$).

model, the kinematics of a unicycle robot to generate velocity commands which are sent as references to the humanoid walking pattern generator.

The proposed method is not intended to cover the whole locomotion ability of a humanoid but rather to provide an efficient locomotion modality when the considered scenario suits the conditions in (Mombaur et al, 2010; Truong et al, 2010), i.e., long distances and uncluttered environments.

An important feature of our method is that we will use only visual information as feedback. As illustrated in Fig. 1, an image processing block elaborates the information coming from the humanoid onboard camera to determine the features used in the visual servoing control scheme based on the unicycle mobility model. The velocity commands issued by this controller represent the desired speed and direction of the humanoid walking motion. These reference velocities are transformed in low-level commands by the robot built-in walking pattern generator.

With respect to the unicycle visual-based control introduced in (Toibero et al, 2009) we prove, using robustness arguments, that it is possible to remove the hypothesis of parallel corridor walls. As an additional original contribution, to allow navigation of the robot in maze-like environments, we propose a strategy that handles the presence of corners and corridor junctions.

3 Visual features model

Consider the wheeled mobile robot with unicycle kinematics shown in Fig. 2. The generalized coordinates (x, y, θ) denote the position and orientation of the robot frame $\mathcal{F}_r$ with respect to the world reference frame $\mathcal{F}_w$ as illustrated in the figure. The kinematic model of this system is readily written as

$$\dot{x} = v \sin \theta$$
$$\dot{y} = v \cos \theta \qquad (1)$$
$$\dot{\theta} = \omega$$

where v and ω are respectively the vehicle driving and steering velocity. In the following, we assume that v is constant and chosen in advance while ω is the input available for control (a common setting for path tracking control of mobile robots).

Assume that a pinhole camera is rigidly mounted on the robot, at a fixed height h with respect to the ground, the optical axis aligned with the robot heading and tilted of an angle equal to γ (see Fig. 2, right). For simplicity, we suppose that the camera focus lies on the vertical axis passing through the origin of the robot frame $\mathcal{F}_r$; as a consequence, the camera moves, as $\mathcal{F}_r$, according to Eq. (1). The controller to be derived is, however, robust with respect to non zero offset in the camera position, as discussed in Sect. 6.

The transformation matrix from the (x_w, y_w) plane, where the points of interest lay, to the camera image plane can be retrieved from the matrix (Ma et al, 2003):

$$\boldsymbol{T}_w^i = \boldsymbol{T}_c^i \cdot \boldsymbol{T}_r^c \cdot \boldsymbol{T}_w^r \qquad (2)$$

where $\boldsymbol{T}_w^r$ expresses the transformation from the world to the robot frame and depends on the robot generalized coordinates only. The transformation $\boldsymbol{T}_r^c$ depends on the camera extrinsic parameters h and γ, while the camera to image transformation $\boldsymbol{T}_c^i$ depends on the camera intrinsic parameters $S_{u,v} = \alpha_{x,y} f$, with f the focal length and α_x / α_y the pixel aspect ratio, and on the principal point coordinates which are set to zero due to the assumption that the optical axis passes through the origin of the image plane. Noticing that the purpose of the controller is to regulate the position of the
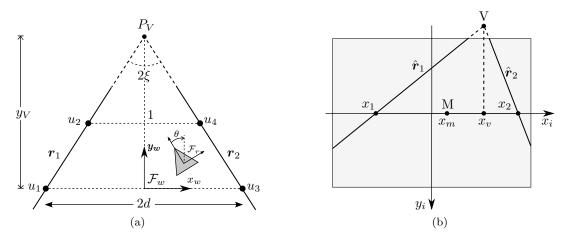
Fig. 3: Representation of the corridor guidelines. In general, they are not parallel and can intersect in the Cartesian space (a). As an effect of the perspective projection, on the image plane (b) the two guidelines always intersect at the vanishing point V; the midpoint of the segment between the intersection points of the guidelines with the abscissa axis defines the middle point M.

robot with respect to the corridor guidelines which are supposed to be on the ground, the $z-$coordinate of the points of interest in the cartesian space are simply set to zero. Hence, by eliminating the third column from $\boldsymbol{T}_w^i$ we obtain the projection matrix for our computations:

$$\boldsymbol{P} = \begin{pmatrix} S_u c_\theta & -S_u s_\theta & -S_u(xc_\theta - ys_\theta) \\ -S_v s_\gamma s_\theta & -S_v s_\gamma c_\theta & S_v s_\gamma(yc_\theta + xs_\theta) + S_v hc_\gamma \\ c_\gamma s_\theta & c_\gamma c_\theta & hs_\gamma - c_\gamma(yc_\theta + xs_\theta) \end{pmatrix},$$
(3)

where the shorthand c_* (s_*) has been used to denote the trigonometric function $\cos(*)$ $(\sin(*))$. Note that the parameters $S_{u,v}$ and h are positive quantities, while the rotation angles are taken as positive according to the direction of the arrows in Fig. 2.

The visual features of interest are given by the images of the edges at the intersection of the corridor walls with the floor which is assumed to be flat. These edges are called *corridor guidelines* and are denoted by $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ in Fig. 3a[1]. The projections of these lines on the image plane, illustrated in Fig. 3b, are denoted respectively by $\hat{\boldsymbol{r}}_1$ and $\hat{\boldsymbol{r}}_2$.

Two useful features can be reconstructed from the visual guidelines $\hat{\boldsymbol{r}}_1$ and $\hat{\boldsymbol{r}}_2$: (*i*) the *vanishing point* V at the intersection of $\hat{\boldsymbol{r}}_1$ and $\hat{\boldsymbol{r}}_2$; (*ii*) the *middle point* M represented by the midpoint of the segment joining the intersection points x_1 and x_2 of $\hat{\boldsymbol{r}}_1$ and $\hat{\boldsymbol{r}}_2$ with the

---

[1] We consider only the case of convergent corridor guidelines with respect to the robot direction of motion. Following corridors with divergent guidelines is limited by the dimension of the camera field of view. The approach proposed in this paper is still valid but the technical details of its application are not discussed here for lack of space.

image plane abscissa axis x_i. In the next sections we derive the visual features equations in the general case of non-parallel corridor guidelines.

3.1 Visual guidelines

The visual features actually used in the unicycle controller are the abscissa coordinates of the vanishing and middle point, respectively denoted by x_v and x_m. To obtain these quantities it is necessary to compute the projection on the image plane of the two corridor guidelines. Placing the world reference frame $\mathcal{F}_w$ on the corridor bisector, the equations of the left and right guidelines, respectively $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, can be defined in $\mathcal{F}_w$ by the points (see Fig. 3a):

$$\boldsymbol{u}_1 = (-d, 0)^T, \ \boldsymbol{u}_2 = (-d + \sigma, 1)^T$$

and

$$\boldsymbol{u}_3 = (d, 0)^T, \ \boldsymbol{u}_4 = (d - \sigma, 1)^T$$

where $\sigma = \tan \xi$, 2ξ is the angle between the two corridor guidelines and $2d$ is the distance between the points $\boldsymbol{u}_1$ and $\boldsymbol{u}_3$ in Fig. 3a at the intersection of x_w with the corridors guidelines. The homogenous representation of the two guidelines is:

$$\tilde{\boldsymbol{r}}_1 = \tilde{\boldsymbol{u}}_1 \times \tilde{\boldsymbol{u}}_2 = (-1, \sigma, -d)^T,$$

$$\tilde{\boldsymbol{r}}_2 = \tilde{\boldsymbol{u}}_3 \times \tilde{\boldsymbol{u}}_4 = (-1, -\sigma, d)^T,$$

where $\tilde{\boldsymbol{u}}_i$ are the homogenous coordinates of the $i-$th point. The images of the corridor guidelines can be easily computed through the projection matrix $\boldsymbol{P}$ (Hartley

and Zisserman, 2004):

$$\hat{\boldsymbol{r}}_1 = \boldsymbol{P}^{-T}\tilde{\boldsymbol{r}}_1 = (a_1, b_1, c_1)^T,$$

$$\hat{\boldsymbol{r}}_2 = \boldsymbol{P}^{-T}\tilde{\boldsymbol{r}}_2 = (a_2, b_2, c_2)^T,$$

(4)

where a_n, b_n and c_n, $n = \{1, 2\}$, represent the coefficients of the visual guidelines expressed in the form $a_n x + b_n y + c_n = 0$.

3.2 Middle point

Given the expression of the visual guidelines in Eq. (4), the abscissa of the middle point is, by definition:

$$x_m = \frac{1}{2}(x_1 + x_2) = -\frac{1}{2}\left(\frac{c_1}{a_1} + \frac{c_2}{a_2}\right),$$

(5)

which can be expressed in terms of the robot generalized coordinates and of the parameters defining the environment and the camera configuration as

$$x_m =$$
$$\frac{k_2 \frac{x}{c_\theta} + k_3 \tan\theta + \sigma\tan\theta[k_3\sigma + k_2\sec\theta(\sigma y - d)]}{1 - \sigma^2\tan^2\theta},$$

(6)

where $k_2 = -S_u s_\gamma / h$, $k_3 = -S_u c_\gamma$.

3.3 Vanishing point

The vanishing point abscissa is determined by the intersection of the visual guidelines $\hat{\boldsymbol{r}}_1$ and $\hat{\boldsymbol{r}}_2$:

$$x_v = \frac{b_1 c_2 - c_1 b_2}{a_1 b_2 - a_2 b_1}$$

(7)

that gives:

$$x_v = \frac{k_1\tan\theta + \sigma k_1 \frac{1}{d}(x - y\tan\theta)}{1 + \sigma\frac{1}{d}(h\tan\gamma\sec\theta - x\tan\theta - y)}$$

(8)

where $k_1 = -S_u / c_\gamma$.

Some comments are now in order.

− Equations (6) and (8) reduce respectively to the expression of the middle and vanishing point abscissae given in Toibero et al (2009) when setting $\sigma = 0$.
− The middle point abscissa x_m depends on both the robot distance to the corridor bisector, encoded in the x coordinate, and on its orientation θ. The dependence on the y coordinate appears, as intuitive, only in case of non-parallel guidelines ($\sigma \neq 0$).
− The vanishing point abscissa x_v depends only on the robot orientation θ when the corridor guidelines are parallel ($\sigma = 0$).

4 Vision-based control

The aim of the controller is to guarantee a stable robot motion along the corridor bisector. Specifically, the robot x and θ coordinates must be stabilized to zero. In both the cases of parallel (Eq. (6) and (8), with $\sigma = 0$) and non-parallel guidelines (Eq. (6) and (8), with $\sigma \neq 0$), this corresponds to zeroing the vanishing and middle point abscissa, as stated in the following.

Proposition 1 *Stabilization to the origin of the image plane of both the middle point and the vanishing point abscissa implies robot navigation along the corridor bisector.*

$$(x_m, x_v)^T \to (0, 0)^T \Rightarrow (x, \theta)^T \to (0, 0)^T.$$

Proof From Eq. (8) it is easy to verify that x_v vanishes in correspondence of all those robot configurations such that $x = \tan\theta(y - y_V)$, where $y_V = d/\sigma$ is the ordinate of the point P_V at the intersection of the corridor guidelines in the $\mathcal{F}_w$ frame (see Fig. 3a). In these configurations the robot heading points to P_V and so does the camera optical axis which is aligned with the robot heading in our setting (see Fig. 2).

Setting $x = \tan\theta(y - y_V)$ in Eq. (6) and solving the equation $x_m = 0$ we find two possible solutions: either $\theta = 0$ (and, hence, both x and θ converge to the desired values) or $(y - y_V) = -\frac{h}{\tan\gamma}\cos\theta$. This last solution corresponds to configurations such that the camera optical axis passes through P_V; in this case, both x and θ could be different from zero and such that the robot position belongs to the arc of circle with center in P_V and radius equal to $r_V = \frac{h}{\tan\gamma}$.

On the image plane this corresponds to $c_1 = c_2 = 0$, i.e., the visual guidelines passing through the origin of the image plane. In this case vanishing and middle point collapse to the origin of the image plane. This point cannot be a stable equilibrium point of the closed loop system since it would imply that either the robot is not moving, but this is not possible since we have assumed that the driving velocity v of the robot is a positive constant, or that it is moving along the circle with center in P_V and radius equal to r_V while pointing to P_V but this is not possible due to the nonholonomic constraint.

In the following Sect. 4.1 we will briefly recall the vision-based control law introduced by Toibero et al (2009) for the nominal case of parallel guidelines. In Sect. 4.2 we will then prove that exponential convergence is preserved also in the case of non-parallel corridor guidelines.

Fig. 4: Robot $x-y$ motion along corridors characterized by different values of the relative wall slope σ and different initial conditions.

4.1 Parallel corridor guidelines

In the case of parallel corridor guidelines, the dynamics of the middle point, expressed as a function of the unicycle driving and steering velocity v and ω, is:

$$\dot{x}_m = \frac{k_2 x_v}{k_1} v + \left(\frac{x_m x_v}{k_1} + k_3 \right) \omega.$$

Considering ω as the input available for control, feedback linearization of the x_m dynamics provides the control law:

$$\omega = \frac{k_1}{k_1 k_3 + x_m x_v} \left(-\frac{k_2}{k_1} v x_v - k_p x_m \right) \quad (9)$$

where k_p is a positive control gain. Toibero et al (2009) proved that this control law stabilizes the equilibrium point $(x_m, x_v) = (0, 0)$ of the visual features dynamics in case of parallel corridor guidelines.

Specifically, the resulting closed-loop dynamics is

$$\dot{x}_m = -k_p x_m \quad (10)$$
$$\dot{x}_v = \frac{-k_1^2 k_2 x_v v - k_1^3 k_p x_m - k_2 x_v^3 v - k_1 k_p x_m x_v^2}{k_1 (k_1 k_3 + x_m x_v)},$$

where the dynamics of x_m is clearly exponentially stable while exponential convergence of x_v has been proven using Lyapunov arguments in Toibero et al (2009).

As an alternative proof, convergence of x_m can be easily shown to imply convergence of x_v due to the non-holonomic constraint.

4.2 Non-parallel corridor guidelines

Using the control law (9) also in the case of non-parallel corridor guidelines the closed-loop dynamics can be written as

$$\dot{x}_m = -k_p x_m + p_m(x_m, x_v)$$
$$\dot{x}_v = f_v(x_m, x_v) + p_v(x_m, x_v) \quad (11)$$

where $p_m(x_m, x_v)$ and $p_v(x_m, x_v)$ are the perturbations to the nominal dynamics (10) induced by the non-null value of $\sigma = \tan \xi$.

In Sect. 9 we show that the terms $p_m(x_m, x_v)$ and $p_v(x_m, x_v)$ are vanishing and locally Lipschitz around the equilibrium $(x_m, x_v)^T = (0, 0)^T$ of the nominal system. This implies that there exists a candidate Lyapunov function guaranteeing that the origin is still an exponentially stable equilibrium point provided that the perturbation satisfies a linear growth bound (Kahlil, 2001)

$$||\boldsymbol{p}|| \leq \delta ||\boldsymbol{x}||,$$

where $\boldsymbol{p} = (p_m \ p_v)^T$ and $\boldsymbol{x} = (x_m \ x_v)^T$, for sufficiently small δ. In Sect. 9 we show that the perturbation terms can be written as $p_m = \sigma \tilde{p}_m$ and $p_v = \sigma \tilde{p}_v$ and then it is possible to write $||\boldsymbol{p}|| = \sigma ||\tilde{\boldsymbol{p}}||$, with $\tilde{\boldsymbol{p}} = (\tilde{p}_m \ \tilde{p}_v)^T$ vanishing and locally Lipschitz around the equilibrium of the nominal system. Therefore by limiting the perturbation σ it is possible to satisfy the sufficient condition of Lemma 9.1 in Kahlil (2001).

However, considering that in general it is not possible to bound σ, because it is a priori unknown, and that the conditions of Lemma 9.1 in Kahlil (2001) are only sufficient and might be too conservative, having been
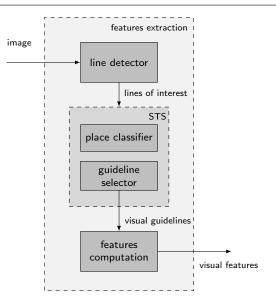
Fig. 5: Feature extraction block diagram.

responding to images generated by the corridor walls) visual guidelines so as to steer the robot in the desired direction. Once the visual guidelines are defined, feature computation follows according to Eq. (5)–(7).

The whole process of feature extraction from the camera images is synthetically illustrated in Fig. 5: the *line detector* extracts from the robot camera images the lines of interest, i.e., the lines candidate to be classified as visual guidelines. A State Transition System (STS) first exploits the output of the line detector to obtain a sort of robot placement classifier. In particular, depending on the number of detected lines, on their slope and position in the image plane, it is possible to determine where the robot is (e.g., along a corridor, in front of a corner, etc.). Based on the robot placement in the environment, the *guideline selector* determines two visual guidelines among the detected lines or artificially defines one or both the guidelines used in the *feature computation* block to determine x_m and x_v.

5.1 Line detector

The *line detector* extracts from images all the non-vertical lines, i.e., the lines whose slope is below a given threshold. The resulting set will, in fact, contain the lines that most likely represent the images of the corridor guidelines (Fig. 6a).

The line detector has a quite standard structure (Laganière, 2011) and its implementation is based on the OpenCV library (OpenCV, 2012). In particular, it uses (*i*) Canny's algorithm for edge detection (white contours in Fig. 6b), (*ii*) probabilistic Hough transform for line segments extraction (Fig. 6c), and (*iii*) a merging procedure to fuse similar segments. This last operation was needed because, due to image noise, the Hough transform can generate many segments actually belonging to the same line. This phenomenon is particularly evident in Fig. 6c where the dots on the blue lines represent the barycenters of the segments found by application of the Hough transform.

The merging procedure can be described by the following steps:

– group the segments issued by the Hough transform (Fig. 6c) based on their slope and y−intercept and discard segments with slopes greater than a threshold (i.e., "vertical" segments);
– for each group of segments, build a mask, i.e., a black-and-white image where the white pixels draw the single line;
– apply the logic conjunction operator between each mask and the Canny output; this provides a set of points for each group of segments, whose interpo-

obtained through worst case analysis, it is of no practical interest to push the analysis further and to determine the interval of σ values for which local asymptotic stability is guaranteed.

The relevant result here is that the relative slope of the corridor guidelines translates in additive perturbations of the x_m and x_v dynamics with respect to which the control law is robust. Nonetheless, we have run simulations for different σ values and different initial conditions. Figure 4 shows that convergence was obtained for a wide range of σ values from almost parallel corridor walls to walls forming a relative angle 2ξ equal to $45°$, with the robot initial condition set to $x_0 = 0.2$ m, $\theta_0 = 0.25$ rad (first three snapshots, from left to right). In the last snapshot the robot motion diverges due to an initial condition ($x_0 = 0.82$ m, $\theta_0 = 0.625$ rad) far from the nominal equilibrium.

5 Navigation in maze-like environments

The control law derived in the previous section allows navigation along corridors enclosed with parallel or non-parallel walls. The feedback signal is provided by the image plane abscissa of the vanishing and middle point, respectively x_v and x_m, computed from the *visual guidelines*, i.e., the images of the corridor guidelines.

A corridor navigation algorithm based on this control implicitly assumes that the visual features are always defined, i.e., that both the corridor guidelines remain in the camera field of view. This is, however, not true when approaching corners and junctions or even for particular starting robot configurations. In these situations it is always possible to define *virtual* (i.e., not cor-

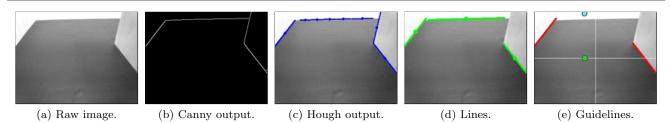| (a) Raw image. | (b) Canny output. | (c) Hough output. | (d) Lines. | (e) Guidelines. |

Fig. 6: The main steps of the image processing. On the gray-scale image caught by the robot camera (a), a Canny algorithm extracts the edges (b), the Hough transform detects the line segments (c), and a merging operation provides the real lines (d). Among these, two guidelines are selected, shown in red in (e). Finally, the vanishing and middle point are computed (the cyan and green circles drawn in (e)).

lation gives the lines of interest (see Fig. 6d) to be sent to the *guideline selector*.

5.2 State Transition System

The procedure described in the previous section selects the lines in the image which are candidate to represent the corridor guidelines.

In the normal straight corridor navigation, the visual features used in the control law (9) are directly computed from the detected lines because, in this case, they correspond to the images of the corridor guidelines. In the presence of corners or junctions, it is necessary to add artificial visual guidelines to steer the robot in the desired direction. To this aim we propose the use of a State Transition System (STS) that extends the corridor following control law (9) to navigation in a network of corridors. In fact, by properly defining the visual guidelines, through the guideline selector, the robot is steered through corridor connections using the same control law with an adjusted feedback signal.

The STS that describes the behavior of the navigation algorithm, is modeled through label functions, states and actions (Baier and Katoen, 2008). The states are used to describe the placement of the robot inside a network of corridors during the navigation. Label functions and actions model the input/output relationships between states.

For illustration purposes, we have defined an STS modeling the robot navigation in a network of corridors connected by corners and T-junctions. The set of states $\{s_i\}$ of the STS is the following:

- $s_1 = empty\ space$. The robot is very close to a corridor wall or its field of view is narrower than the corridor. In this case no line segment can be detected and the robot has free space on both the right and left side (see Fig. 7a).
- $s_2 = facing\ a\ wall$. The robot is pointing toward a wall either because it is not aligned with the cor-

ridor bisector or because it is at a corner or at a T-junction. Typically, the camera can detect just one line segment (see Fig. 7c).
- $s_3 = corridor$. The robot is along a straight corridor and the two corridor guidelines are visible in the camera image plane (see Fig. 7e).
- $s_4 = dangerous\ turn$. This state models the ambiguous situation that occurs when the robot is not aligned with the corridor bisector close to a turn. The two detected lines are not the correct visual guidelines since they are the images of guidelines belonging to two different corridor segments (see Fig. 7g). In this case, the two detected line segments do not intersect.
- $s_5 = facing\ an\ edge$. The robot is in front of an edge and, similarly to s_4, the two detected lines do not correspond to the correct visual guidelines (see Fig. 7i). Differently from s_4, the two line segments intersect within the image plane.
- $s_6 = approaching\ a\ turn$. The robot is close to a curve or a T-junction and three line segments are detected (see Fig. 7k).

Each of the above states can be an initial state, i.e., the algorithm can start with the robot placed everywhere inside the maze.

In the designed STS, the set of actions $\{a_i\}$ is used to model the states output. These actions correspond to the appropriate definition of the two visual guidelines according to the robot position inside the maze. Indeed, the actions of the STS implement the *guideline selector*: according to the particular state (position of the robot inside the maze), a proper action is taken, i.e., two visual guidelines are appropriately defined. Note once again that the visual guidelines used to compute the middle and vanishing points in Eq. (9) can be directly defined by the image of the corridor guidelines, or by ad hoc defined artificial lines. The STS possible actions are:

- $a_1 = add\ two\ artificial\ guidelines\ respectively\ on\ the\ left\ and\ right\ side\ of\ the\ image\ plane$ (see Fig. 7d).

(a) s_1: empty space

(b) a_2 action

(c) s_2: facing a wall

(d) a_1 action

(e) s_3: corridor

(f) a_4 action

(g) s_4: dangerous turn

(h) a_3 action

(i) s_5: facing an edge

(j) a_3 action

(k) s_6: approaching a turn
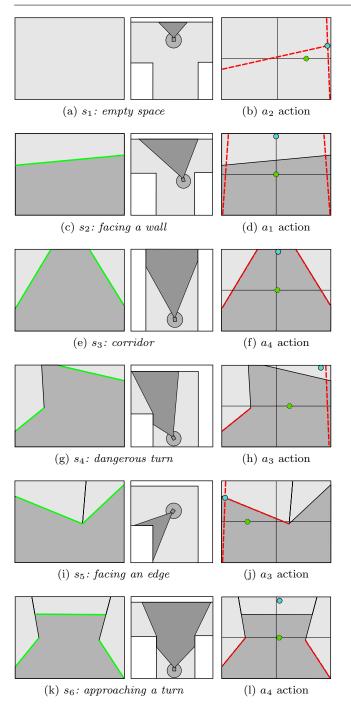
(l) a_4 action

Fig. 7: States and actions of the STS used by the *guideline selector* during the navigation in corridors connected by turns and T-junctions. At the center, the top view of the robot placements in the environment with the camera field of view highlighted in gray shadows. On the left, the corresponding camera views with the typically detected lines in bold green. On the right, the selected visual guidelines (dashed red) and the corresponding visual features (x_m is the green disk with dashed boundary, x_v is the cyan disk with continuous boundary).

This suggests the robot to proceed straight because there is just one line with slope below a given threshold and intercept close to the superior border of the image plane. Typically this happens when the robot is close to a turn, but it is not yet the time to steer.

- a_2 = *add two artificial guidelines, one vertical and one oblique* (see Fig. 7b). The introduction of such visual guidelines forces the robot to turn and is used when there are no detected lines or when the only line on the image plane has a slope below a given threshold and intercept close to the inferior border of the image plane.

- a_3 = *select as guideline one of the detected segments in the image plane and add a vertical artificial one.* This action is used to manage the ambiguous situations described by states s_4 and s_5 (see Fig. 7h–7j) or when just one oblique line is detected.

- a_4 = *select the two detected oblique lines as guidelines.* Both the corridor guidelines are detected and chosen as guidelines (like in the situations depicted in Figures 7f and 7l).

To complete the STS definition, it is necessary to model the state inputs, i.e., the conditions according to which the robot placement inside the maze can be classified as one of the states s_i. To this end, we use the labeling function $L(s_i)$ that relates the set of states $\{s_i\}$ to a set of atomic propositions $\{p_i\}$, i.e., proposition that can be simply false or true and do not depend on any other proposition.

We defined the following atomic propositions evaluated on the output of the line detector:

p_1 = *no line segment is detected*
p_2 = *one line segment is detected*
p_3 = *two line segments are detected*
p_4 = *three line segments are detected*
p_5 = *the lines supporting the detected segments intersect outside the image plane*
p_6 = *the detected line segments intersect*

As an example, proposition p_5 is true in the situation depicted in Fig. 7e, while Fig. 7i reports a case in which p_6 is true.

$L(s_i)$ is the set of atomic propositions that are true at the state s_i. The following labeling functions have been defined for the designed STS:

$$
\begin{aligned}
L(s_1) &= L(empty\ space) &&= \{p_1\} \\
L(s_2) &= L(facing\ a\ wall) &&= \{p_2\} \\
L(s_3) &= L(corridor) &&= \{p_3, p_5\} \\
L(s_4) &= L(dangerous\ turn) &&= \{p_3, \bar{p}_5, \bar{p}_6\} \\
L(s_5) &= L(facing\ an\ edge) &&= \{p_3, \bar{p}_5, p_6\} \\
L(s_6) &= L(approaching\ a\ turn) &&= \{p_4\}.
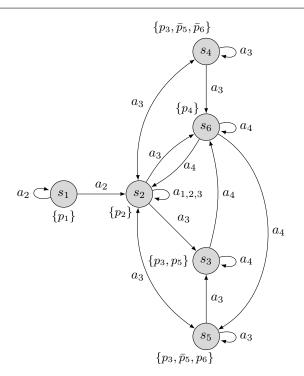\end{aligned}
$$

Fig. 8: The STS describing the generation of the feedback signal through the appropriate definition of the visual guidelines. A place classifier defines the current state of the robot while a guideline selector actually defines the action to be taken by defining the visual guidelines.

A bar over the variables denotes the negation of the corresponding atomic proposition.

The STS is graphically illustrated in Fig. 8. Gray circles represent the states s_i, while transitions between states are illustrated through black arrows. The output of the state s_i is modeled by the action a_i, while the input is determined by the labeling function $L(s_i)$.

Note that the STS guarantees a deterministic behavior provided that the assumptions of the method are satisfied, i.e., flat ground, corridor-like environment with curves and T-junctions. In particular, the open-loop phase that could occur at the transition from one state to the other ends as soon as a corridor guideline enters the camera field of view (if this does not happen, then the robot is navigating in the free space). At this point the visual feedback drives the robot away from the wall, thus guaranteeing its safety.

5.3 Feature computation

Once selected the two guidelines, the vanishing and the middle point are easily computed: x_m is the midpoint of the segment connecting the intersections of the two vi-
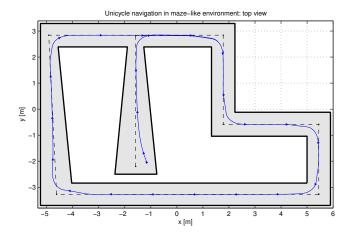

Fig. 9: Simulation results of the corridor navigation algorithm in a maze-like environment.

sual guidelines with the image plane abscissa axis given by Eq. (5), while x_v is the image plane abscissa of the intersection point between the visual guidelines and it is computed through (7). Figure 6e shows the vanishing and the middle point with cyan (continuous boundary) and green (dashed boundary) circles respectively. For visualization, the value of the vanishing point ordinate has been saturated to the border of the image plane.

5.4 Simulation results

To validate the effectiveness of the whole navigation algorithm we have run dynamic simulations with a unicycle robot in a network of corridors connected by corners and T-junctions. A 8 Hz low-pass filter smooths the features discontinuities due to the introduction of the virtual guidelines by the STS. The filtered signals are used to compute the control action (9) at 10 Hz.

Figure 9 reports the 2D map of the simulated environment with the plot of the path traveled by the unicycle robot (blu continuous line). The black dashed lines represent the corridor segments bisector. The exponential convergence of the unicycle to the corridor bisector can be appreciated in correspondence of long enough corridor segments. The accompanying video shows the 3D robot motion within the Cyberbotics-Webots (2012) dynamic simulation environment.

6 From unicycles to humanoids

The unicycle dynamics (1) represent an abstract mathematical model of mobility. For wheeled mobile robots, this model accounts for the nonholonomic constraints deriving from the rolling wheels; i.e., it has a physical

motivation due to the mechanical nature of the motion generation system. In real robots, however, the actual control inputs could be different from the velocity commands v and ω in (1). For differential-drive vehicles, for example, these commands must be transformed to rotational speed for the wheels. By analogy, we need to convert the unicycle feedback commands in admissible inputs for the locomotion system of a humanoid.

Humanoid robots are endowed with omnidirectional walk capability. The mobility model (1) is therefore admissible for these systems and appropriate in long distance navigation (Mombaur et al, 2010; Truong et al, 2010). The velocity commands v and ω generated for the unicycle control can be converted in admissible inputs for the low-level locomotion controller so that the humanoid follows the same path as the unicycle.

For the humanoid robot NAO, adopted as experimental platform, this paradigm can be instantiated by using the built-in method *move* allowing direct specification of the driving and steering velocity of the robot. Since the most recent command overrides all previous commands, this function can be called with arbitrary frequency, thus providing the reactive behavior required by visual-based control.

The input arguments sent to *move* are the constant driving velocity v of the virtual unicycle and the steering velocity ω determined through Eq. (9). These commands are translated by a low-level walking pattern generator in desired step length, direction and frequency. Alternatively, it is possible to directly control these gait parameters using the function *setWalkTargetVelocity*, as done in Faragasso et al (2013). The two methods are equivalent from the navigation performance point of view but the second requires additional computations to transform the reference velocities v and ω in walking gait parameters. Using *move* these computations are done very efficiently by the built-in walking motion generator.

The complete control scheme is illustrated in Fig. 10. The visual information provided by the robot onboard camera is used in the feedback control law (9) designed on the nominal system. In Sect. 4.2 this controller has, in fact, be shown to be robust with respect to the perturbation induced by non-parallel corridor walls on the nominal closed-loop dynamics.

The control action is computed at the average frequency of 10 Hz using as feedback signals the vanishing and middle point obtained from the robot camera images as illustrated in Sect. 5. The raw visual information is affected by the the sway motion due to walking, an event assessed at 1 Hz. Following the frequency-based approach to sway motion cancellation illustrated in (Oriolo et al, 2013), a low-pass filter with cutoff
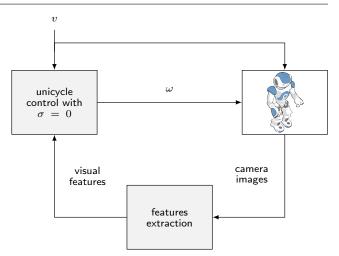

Fig. 10: The control loop for the humanoid robot NAO. The block *features extraction* contains a STS for maze navigation and a low-pass filter to smooth the feedback signal and cancel the oscillatory motion due to walking and the discontinuities.

frequency of 0.8 Hz is used in the line detection process to mitigate the effect of walking on the visual features motion. The experimental results in (Oriolo et al, 2013) show that this approach is more effective than a kinematic-based cancellation in the case of NAO. Our approach can be considered as a decoupled method like, e.g., (Dune et al, 2010) in which the sway motion is isolated and used to correct the measured features motion by computing the difference between the walking pattern generator reference velocity and the real velocity of the robot. Methods like (Garcia et al, 2015), embedding the visual servoing in the walking pattern generator could not be applied in the case of NAO because access to the walking pattern generator is not granted by the built-in control software.

This filter used in the line detection process, however, is not effective in smoothening the features discontinuities due to the introduction of the virtual guidelines, because these lines are artificially defined by the STS when a transition is necessary. As for the simulations reported in Sect. 5.4, a 8 Hz low-pass filter was necessary to eliminate the discontinuities due to the transitions at curves and junctions. This filter does not alter the output of the line detection algorithm while making the feedback signal provided by the visual features motion smoother.

To show the nature of the signals involved in the navigation process and the effect of the filtering action, we report in Fig. 11 the features evolution and the control action from an experiment in which NAO

negotiates a left curve. The plots of Fig. 11 report both the raw (black dashed lines) data and the signal obtained by filtering them (red continuous line) with a 8 Hz low-pass filter.

For the first 27 seconds, the robot navigates in a straight corridor, and the visual features motion is kept close to zero by the control action. At about 27 s, the STS adds a left artificial guideline because the robot is approaching the curve and the real left guideline disappears from the camera image plane. As a result, the vanishing point rapidly moves towards the left border of the image plane, this corresponds to the negative step in Fig. 11(b). Around 40 s, also the right guideline disappears for few seconds and the STS correspondingly adds a right artificial guidelines (positive peak at ~40 s). The measurement noise induces some oscillations in the features motion until the corridor guideline in front of the robot becomes of interest for the STS that recovers it as the guideline allowing the negotiation of the left curve at about 43 s. After recovering the overshoot between 53 s and 60 s, the robot recovers the corridor bisector.

Note how the filtering action allows to smooth the peaks of the signals due to the transitions between the STS states and also to reduce the measurement noise. As a result, also the angular velocity command does not present excessive peaks. It is, however, worth emphasizing that the commanded angular velocity is naturally "filtered" by the robot slow dynamics and the peaks due to noise have actually effect on the robot motion at a different scale as it might appear in the time plots. The live feature evolution on the image plane shown in the video accompanying the paper is, in fact, more informative of the actual robot motion. Both the feature evolution (directly) and the control action (indirectly) can be better appreciated from these clips.

Finally, it is worth noticing that through the function *move* it is possible to control the path of a point between the feet and the position and orientation of the feet along the averaged (i.e, after sway motion cancellation) path of this point. Using our mobility model, the feet are always oriented along the path tangent. Due to its kinematics NAO's torso follows the orientation of the feet with reasonable approximation and so does the camera placed in the robot head if the neck yaw is kept in its zero configuration, i.e., the head oriented as the torso. The error due to the approximation mentioned above is corrected by the feedback action, as confirmed by our experiments.

It should also be noted that, due to its position, the motion of NAO's camera, after cancellation of the sway motion, does not correspond exactly to the mobility model expressed by Eq. (1). The camera focus is, in fact, slightly displaced, along the sagittal plane, with respect to the point controlled through the function *move*. This results in a vanishing perturbation of the features dynamics the analysis of which is analogous to that induced by the non-zero relative inclination of the corridor walls and is omitted for lack of space. Note however that, for a more realistic model of mobility, the unicycle simulations have been run with a camera placed at a distance of about 15 cm from the wheel axis midpoint along the robot sagittal plane.

7 Experimental results

The proposed visual navigation controller has been experimentally validated on the robot NAO. This small-sized humanoid is equipped with a CMOS camera with a 72.6° diagonal field of view mounted on its forehead. The camera intrinsic parameters have been determined using the MATLAB calibration toolbox. In the experiments, the head does not move with respect to the torso and its pitch angle has been set to 29.5°. Considering that the focal axis is slightly tilted (1.2°) with respect the forward axis of the head, we have set $\gamma = 30.7°$ in the control law (9). The measure of the distance between the camera and the feet, with the robot at the walking posture configuration, has given $h = 0.53$ m.

The video stream has been captured with a resolution of 320×240 pixels and sent to a remote computer (3.4 GHz Intel i7 processor and 8 GB RAM), through Wi-Fi connectivity, to perform the image processing detailed in Section 5 at a frequency of 10 Hz (the control runs at the same rate). The driving velocity has been set to $v = 0.055$ m/s and the proportional gain in the ω expression has been chosen as $k_p = 0.75$.

The presented experiments have been performed in environments tailored to the size of NAO. In the first experiment, NAO starts walking off the center of a straight corridor enclosed with non-parallel walls. Figure 12 reports four snapshots from the video accompanying the paper and showing NAO's convergence to the corridor center. The convergence to the corridor bisector can be appreciated from the movie clip accompanying the paper, also in the case of corridor with parallel walls.

The second and third experiments aim at assessing the effectiveness of the guidelines selector approach in handling the absence of real visual guidelines to drive the robot in presence of corners and junction. Snapshots of the second experiment (Fig. 13) show NAO negotiating a left turn. According to the STS described in Sect. 5.2, NAO correctly approaches the turn (first snapshot), faces the corner (second snapshot) and turns

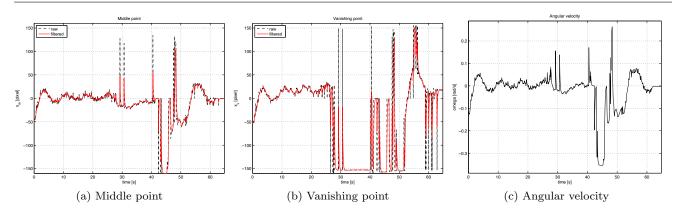| (a) Middle point | (b) Vanishing point | (c) Angular velocity |

Fig. 11: NAO turning at a left curve: time evolution of the middle (a) and vanishing point (b), and the computed angular velocity (c).
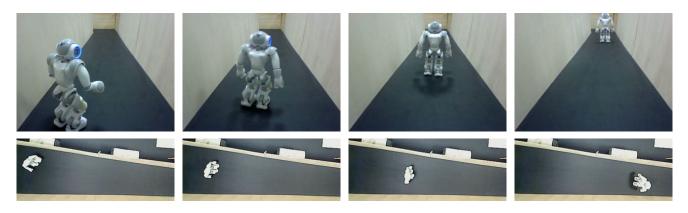

Fig. 12: First experiment: vision-based corridor navigation with off-centered start. The first two snapshots show NAO starting away from the corridor center but rapidly recovering it. The last two snapshots illustrate how NAO is able to keep walking at the center of the corridor.

Fig. 13: Second experiment: negotiating a turn. In the first snapshot NAO is approaching a left turn. The subsequent snapshots show how the robot correctly detects the corner and keeps the center of the corridor also during the turn.

(third snapshot); finally, it restores the corridor navigation (last snapshot). Figure 14 shows NAO keeping the center of the corridor after turning right at a T-junction.

The last experiment shows the robustness of the proposed navigation algorithm. NAO continuously walks for 15 minutes in a closed track, composed of four corridors and corners. Four snapshots of this endurance navigation experiment are shown in Fig. 15.

All the experiments are shown integrally in the video accompanying this paper. In particular, the live video feed from the robot camera with the reconstructed x_v and x_m is included in the clips.

8 Conclusions

We have presented a visual control approach for humanoid indoor navigation. In particular, the control

Fig. 14: Third experiment: turning at a T-junction. The first snapshot shows NAO approaching the junction. When the junction is detected the robot takes the specified direction (second and third snapshot) and resumes walking at the center of the corridor (fourth snapshot).

Fig. 15: Endurance navigation experiment: NAO walks for 15 minutes in a closed track.

objective is to follow a corridor by walking as close as possible to its center and executing safe turns at corners and junctions.

The proposed algorithm is based on a controller for corridor navigation designed for unicycle-like mobile robots, which was extended to effectively handle the presence of turns and junctions without changing the control strategy. Assuming that the robot is cruising at constant speed, the controller provides the angular velocity that corrects its position and orientation based on the visual feedback. The feedback signal is then transformed into inputs that are compatible with the locomotion system of the humanoid producing a natural, human-like walking behavior.

A State Transition System has been implemented to experimentally prove the validity of the approach for the navigation of NAO in a network of corridors connected by curves and T-junctions.

An additional contribution of this work was to show that the exponential convergence of the vision-based control law is preserved also in case of non-parallel corridor guidelines provided that the relative slope of the guidelines is within given bounds. For the reasons discussed in Sect. 4.2 the derivation of these bounds has a very limited practical interest and, therefore, the algorithm was rather tested for a wide range of guidelines inclinations compatible with the camera field of view.

Future work will include:

- quantitative assessment of performance in terms of actual robot motion using a motion capture system currently not available at our lab;
- validation of the controller on real-sized humanoid moving in full-scale maze-like environments;
- avoidance of obstacles possibly present in the corridor;
- place recognition and categorization for other kind of intersections.

9 Appendix

In this section we analyze the perturbtion terms in Eq. (11) to prove that they are vanishing and locally Lipschitz around the equilibrium point $(x_m, x_v) = (0, 0)$.

Computing the time derivative of the visual features expressions (6) and (8) in the case of non-parallel corridor walls and using the unicycle model (1) and the control designed on the nominal system (9) we obtain, for the middle point x_m the closed-loop dymanics

$$\dot{x}_m = -k_p x_m + p_m = -k_p x_m + A_m + B_m + C_m + D_m,$$

where the perturbation p_m is composed by the terms

$$A_m = \sigma \left(x_m \omega + k_2 v\right) \frac{1}{dk_1} \frac{x_v(h \tan \gamma \sec \theta - (x_v/k_1)x) - k_1 x}{1 - (\sigma/d)(y - (x_v/k_1)x)},$$
$$B_m = E_m \tan^2 \theta (k_3 \omega + k_2 v \tan \theta),$$
$$C_m = E_m \tan \theta \left(2 x_m \sec^2 \theta \omega + k_2 v\right),$$
$$D_m = E_m \theta \left(k_3 + k_2 \sec^3 \theta (y - d/\sigma)\right) \omega,$$

(12)

with

$$E_m = \frac{\sigma^2}{1 - \sigma^2 \tan^2 \theta}.$$

To show that the perturbation is vanishing we should invert the map $(x, \theta) \rightarrow (x_m, x_v)$ so as to express the perturbation term as a function of (x_m, x_v) only. However, in Proposition 1 we have shown that at the equilibrium $(x_m, x_v)^T = (0, 0)^T \Rightarrow (x, \theta)^T = (0, 0)^T$ except on points of the circle with center P_V, the intersection point of the two corridor guidelines, and radius $r_V = \frac{h}{\tan \gamma}$.

These points, however, cannot be stable equilibria of the closed loop dynamics, as also shown in Proposition 1. In addition, considering the robot footprint with respect to the corridor width, these points can hardly be reached in practical situations. Hence, by ignoring these points, we can evaluate the perturbation terms at the equilibrium of the nominal system by setting $(x_m, x_v) = (0, 0)$ and $(x, \theta) = (0, 0)$.

A quick inspection of Eq. (12) provides evidence that p_m is null at the equilibrium point $(x_m, x_v) = (0, 0)$, implying that the perturbation induced by the non-parallel corridor guidelines on the closed loop nominal dynamics is non-persistent.

The perturbation term is composed by sums and products of functions that are locally Lipschitz around the equilibrium of the nominal system with the exception of the terms A_m and E_m which present, respectively, the singularities analyzed below.

- $y - y_V = \frac{x_v}{k_1} x$: this singularity cannot be met around the equilibrium point since this would imply that the robot is very close to the point P_V at the intersection point of the corridor guidelines, a situation physically impossible;
- $\tan \theta = \pm 1/\sigma$: is verified if the robot heading is perpendicular to the corridor walls, a situation not possible around the origin of the system if $\sigma \neq 0$.

Finally, given the terms in (12), p_m can be easily expressed as $p_m = \sigma \tilde{p}_m$.

To prove that the perturbation of the vanishing point closed-loop dynamics is non-persisting and locally Lipstchiz at the equilibrium of the nominal system we proceed in a way analogous to the case of the middle point. The closed loop dynamics of x_v in case of non-parallel corridors guidelines is

$$\dot{x}_v = f_v(x_m, x_v) + D_v + E_v = f_v(x_m, x_v) + p_v,$$

where $f_v(x_m, x_v)$ represents the nominal closed-loop dynamics in Eq. (10), while the perturbation p_v is com-

posed of the following terms

$$E_v = \sigma^2 \frac{1}{k_1 d^2} \left(\frac{x_v(h \tan \gamma \sec \theta - (x_v/k_1)x) - k_1 x}{1 - (\sigma/d)(y - (x_v/k_1)x)} \right)^2 \omega$$
$$D_v = \frac{(\dot{B}_v - \dot{A}_v C_v)(1 + C_v) - \dot{C}_v(A_v + B_v)}{(1 + C_v)^2}$$

where

$$A_v = k_1 \tan \theta,$$

$$\dot{A}_v = k_1(\tan^2 \theta + 1)\omega,$$

$$B_v = \sigma k_1 \frac{1}{d}(x - y \tan \theta),$$

$$\dot{B}_v = -\frac{\sigma}{d} y \dot{A}_v,$$

$$C_v = \sigma \frac{1}{d}(h \tan \gamma \sec \theta - x \tan \theta - y),$$

$$\dot{C}_v = \sigma \frac{1}{d} \sec \theta ((h \tan \gamma s_\theta - x) \sec \theta \omega - v).$$

As in the previous case, we can prove that the perturbation vanishes if we set $(x_m, x_v) = (0, 0)$ and $(x, \theta) = (0, 0)$. Also in this case, the perturbation term has some singularities. In particular, the term E_v presents the same singularity as A_m in Eq. (12) discussed above. The term $1 + C_v$ at the denominator of D_v is null if the following equation is verified

$$(y_V - y) \cos \theta - x \sin \theta = -h \tan \gamma.$$

The simple geometric construction in Fig. 16 shows that the term on the left represents the signed distance of the robot to the line orthogonal to the optical axis (directed as the robot heading) and passing through the point P_V where the corridor guidelines intersects. This distance becomes negative, and eventually equal to $h \tan \gamma$, if the robot crosses this line. Around the equilibrium point this would mean that the robot would intersect this line close to P_V, again a non-operative condition.

We can then state that around the equilibrium the perturbation term p_v is locally Lipschitz being given by sums and products of locally Lipschitz functions. Analogously to the perturbation of the nominal middle point dynamics, p_v can be written as $p_v = \sigma \tilde{p}_v$.

Wrapping up, the perturbation terms generated by the non-parallel wall corridor condition have been shown to be vanishing and locally Lipschitz around the equilibrium of the nominal dynamics. Furthermore, the perturbation term is proportional to the perturbation parameter σ representing the corridor walls relative slope.
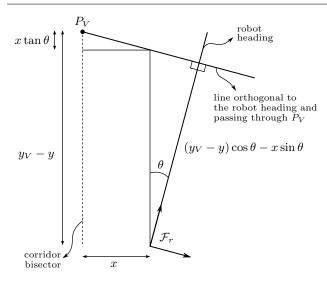
Fig. 16: Geometric interpretation of the D_v singularity.

References

Back I, Kallio J, Makela K (2012) Enhanced map-based indoor navigation system of a humanoid robot using ultrasound measurements. Intelligent Control and Automation 3(2):111–116

Baier C, Katoen JP (2008) Principles of Model Checking. MIT Press, Cambridge, MA

Cherubini A, Chaumette F, Oriolo G (2008) An image-based visual servoing scheme for following paths with nonholonomic mobile robots. In: 10th IEEE International Conference on Control, Automation, Robotics and Vision, pp 108–113

Courty N, Marchand E, Arnaldi B (2001) Through-the-eyes control of a virtual humanoid. In: The Fourteenth Conference on Computer Animation, pp 74–83

Cyberbotics-Webots (2012) Webots: robot simulation software. URL http://www.cyberbotics.com

Delfin J, Becerra H, Arechavaleta G (2014) Visual path following using a sequence of target images and smooth robot velocities for humanoid navigation. In: 2014 14th IEEE-RAS International Conference on Humanoid Robots, pp 354–359

Delgado-Galvan J, Navarro-Ramirez A, Nunez-Varela J, Puente-Montejano C, Martinez-Perez F (2015) Vision-based humanoid robot navigation in a featureless environment. In: Carrasco-Ochoa JA, Martínez-Trinidad JF, Sossa-Azuela JH, Olvera López JA, Famili F (eds) Pattern Recognition, Lecture Notes in Computer Science, vol 9116, Springer International Publishing, pp 169–178

Dune C, Herdt A, Stasse O, Wieber PB, Yokoi K, Yoshida E (2010) Cancelling the sway motion of dynamic walking in visual servoing. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 3175–3180

Dune C, Herdt A, Marchand E, Stasse O, Wieber PB, Yoshida E (2011) Vision based control for humanoid robots. In: 2011 IROS Workshop on Visual Control of Mobile Robots, pp 19–26

Faragasso A, Oriolo G, Paolillo A, Vendittelli M (2013) Vision-based corridor navigation for humanoid robots. In: 2013 IEEE International Conference on Robotics and Automation, pp 3190–3195

Garcia M, Stasse O, Hayet JB (2014) Vision-driven walking pattern generation for humanoid reactive walking. In: 2014 IEEE International Conference on Robotics and Automation, pp 216–221

Garcia M, Stasse O, Hayet JB, Dune C, Esteves C, Laumond JP (2015) Vision-guided motion primitives for humanoid reactive walking: Decoupled versus coupled approaches. The International Journal of Robotics Research 34(4-5):402–419

George L, Mazel A (2013) Humanoid robot indoor navigation based on 2D bar codes: Application to the NAO robot. In: 13th IEEE-RAS International Conference on Humanoid Robots

Hartley RI, Zisserman A (2004) Multiple View Geometry in Computer Vision. Cambridge University Press

Ido J, Shimizu Y, Matsumoto Y, Ogasawara T (2009) Indoor navigation for a humanoid robot using a view sequence. The International Journal of Robotics Research 28(2):315–325

Kahlil HK (2001) Nonlinear systems. Prentice Hall

Kuffner Jr JJ, Nishiwaki K, Kagami S, Inaba M, Inoue H (2001) Footstep planning among obstacles for biped robots. In: 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 500–505

Laganière R (2011) OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 recipes to master this library of programming functions for real-time computer vision. Packt Publishing Ltd

Lutz C, Atmanspacher F, Hornung A, Bennewitz M (2012) Nao walking down a ramp autonomously. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 5169–5170

Ma Y, Soatto S, Kosecka J, Sastry SS (2003) An Invitation to 3-D Vision: From Images to Geometric Models. SpringerVerlag

Maier D, Hornung A, Bennewitz M (2012) Real-time navigation in 3D environments based on depth camera data. In: 2012 12th IEEE-RAS International Conference on Humanoid Robots, pp 692–697

Maier D, Stachniss C, Bennewitz M (2013) Vision-based humanoid navigation using self-supervised obstacle detection. International Journal of Humanoid

Robotics 10(02):1350016

Matsumoto Y, Ikeda K, Inaba M, Inoue H (2000) Exploration and navigation in corridor environment based on omni-view sequence. In: 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1505–1510

Mombaur K, Truong A, Laumond JP (2010) From human to humanoid locomotion – an inverse optimal control approach. Autonomous Robots 28:369–383

Ohnishi N, Imiya A (2013) Appearance-based navigation and homing for autonomous mobile robot. Image and Vision Computing 31(6):511–532

OpenCV (2012) Open source computer vision. URL http://opencv.org/

Oriolo G, Paolillo A, Rosa L, Vendittelli M (2013) Vision-based trajectory control for humanoid navigation. In: 2013 IEEE-RAS International Conference on Humanoid Robots, pp 118–123

Park Y, Suh H (2010) Predictive visual recognition of types of structural corridor landmarks for mobile robot navigation. In: 19th IEEE International Symposium on Robot and Human Interactive Communication, pp 391–396

Sabe K, Fukuchi M, Gutmann JS, Ohashi T, Kawamoto K, Yoshigahara T (2004) Obstacle avoidance and path planning for humanoid robots using stereo vision. In: 2004 IEEE International Conference on Robotics and Automation, pp 592–597

Tello Gamarra D, Bastos-Filho T, Sarcinelli-Filho M (2005) Controlling the navigation of a mobile robot in a corridor with redundant controllers. In: 2005 IEEE International Conference on Robotics and Automation, pp 3844–3849

Toibero JM, Soria CM, Roberti F, Carelli R, Fiorini P (2009) Switching visual servoing approach for stable corridor navigation. In: 14th International Conference on Advanced Robotics, pp 1–6

Truong TVA, Flavigne D, Pettre J, Mombaur K, , Laumond JP (2010) Reactive synthesizing of human locomotion combining nonholonomic and holonomic behaviors. In: 3rd IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, pp 632–637

Vassallo R, Schneebeli HJ, Santos-Victor J (2000) Visual servoing and appearance for navigation. Robotics and Autonomous Systems 31(1–2):87–97

Wei C, Xu J, Wang C, Wiggers P, Hindriks K (2014) An approach to navigation for the humanoid robot nao in domestic environments. In: Natraj A, Cameron S, Melhuish C, Witkowski M (eds) Towards Autonomous Robotic Systems, Lecture Notes in Computer Science, vol 8069, Springer Berlin Heidelberg, pp 298–310