J Math Imaging Vis (2012) 42:196–211
DOI 10.1007/s10851-011-0286-y

Visual Servoing of Legged Robots

**Z. Echegoyen · J.M. Lopez-Guede ·
B. Fernandez-Gauna · M. Graña**

Published online: 29 April 2011

Abstract We build and test a Visual Servoing for all degrees of freedom of a legged robot. We provide a detailed geometrical description relevant to the construction of the Jacobian matrix containing the dependencies of the visual features on the robot joint angles. This matrix embodies the forward kinematics model. To obtain an autonomous control system invariant to world position, we define the ground reference system relative to the basic support points. The control of the robot is computed by the inversion of the forward kinematics model, with two corrections. First, to preserve the ground reference system we must correct the motion of the supporting points. Second, we test a stability condition to avoid the robot to move into unstable configurations. We have tested the approach on a controlled environment to assess its real life performance. The experimental results show the robustness of the approach.

Keywords Visual Servoing · Legged robots · Aibo

1 Introduction

Visual Servoing [6, 16, 21, 28, 29, 32, 33, 36, 37] is defined as the task of positioning one or more robots in order to reach specific poses of their final effectors using, in the closed control loop feedback, the estimated position error computed from the visual information extracted from the environment by one or more video-cameras. Figure 1 illustrates the main feedback loop in image-based Visual Servoing.

Legged robots have been the subject of study for a while [9, 26]. The hexapod and the four legged robots are the most common types. Among them the Sony's Aibo robot has been a privileged research platform [3, 22, 35]. The control and generation of gaits has been a subject of extensive research [1, 13, 17, 18, 23, 24, 38]. General works on Visual servoing for legged robots deal only with the effectors linked directly to the camera [4, 19, 20]. Specifically for the RoboCup robot soccer competition some rough Visual Servoing approaches [25, 30] have been implemented in the Aibo robot to track the ball. However, these approaches are limited to the movement of the head effectors in order to keep the ball inside the robot camera field of view. The space in which the ball can be followed is restricted by the robot's body pose.

In the work reported in this paper we build a forward kinematics model that relates the visual features to all the robot degrees of freedom. Inversion of this model allows to perform Visual Servoing changing the robot's body pose. We detail the construction of the Jacobian matrix relating the dependencies of the visual features on all the robot joint angles. Some initial works were presented in [10, 11].

To obtain an autonomous control system invariant to world position and orientation of the robot we define a ground reference system on the robot supporting points. The forward kinematic model is built over this reference system. The computation of the joint control commands through the inversion of the kinematics model is constrained in order to ensure that the supporting points will remain fixed, to preserve the ground reference system. Besides, there is a condition for stability of the robot's pose [2] that must be checked before applying the computed control commands. This condition consists in the location of the robot's center of mass projection inside the convex hull defined by the robot supporting points.

We have performed real life experiments on a Sony's Aibo ERS-7 robot under controlled conditions to asses the

Z. Echegoyen (✉) · J.M. Lopez-Guede · B. Fernandez-Gauna ·
M. Graña
Grupo Inteligencia Computacional, UPV/EHU, Donostia, Spain
e-mail: zelmar.echegoyen@gmail.com

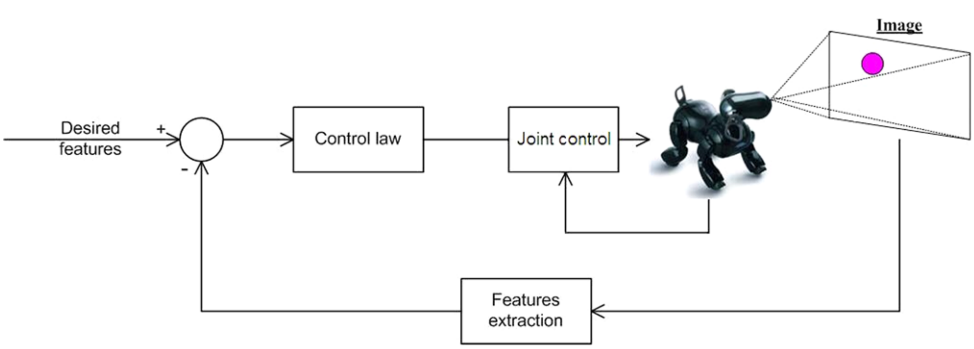


Fig. 1 Visual Servoing feedback loop

Table 1 Notation used across the article

I_c, I_b, I_g	Coordinate Reference Systems for the camera, body and ground
${}^g\mathbf{s}, {}^c\mathbf{s}$	Feature point **s** expressed in the Coordinate Reference System I_g, I_c
${}_iI_j$	Transformation between Coordinate Reference Systems
$\boldsymbol{\theta}_l, \boldsymbol{\theta}_u$	Angles of the leg and upper body joints
$\mathbf{G}_\pi, \mathbf{G}_e$	Basic and extended sets of support points
$\mathbf{p}$	The extended support points and the upper body joints
$J_{s\theta}$	Dependence of image features on the joint angles
J_{sp}	Dependence of image features on **p**
J_{sc}	Dependence of image features on the camera
$J_{p\theta}$	Dependence of **p** on the joint angles
$J_{c\theta_u}$	Dependence of the camera on the upper body joing angles
J_{ce}	Dependence of the camera on $\mathbf{G}_e$
$J_{c\pi}$	Dependence of the camera on $\mathbf{G}_\pi$
$J_{\pi e}$	Dependence of $\mathbf{G}_\pi$ on $\mathbf{G}_e$
$J_{e\theta}$	Dependence of $\mathbf{G}_e$ on the joint angles

applicability of our approach, reporting the quantitative results of such experiments. The visual feature is the image center of the detected region corresponding to the target ball. The degrees of freedom affected by the visual servoing are all the leg's articulations, as well as the neck and head articulations. We have found that the approach provide high accuracy in the positioning of the visual feature (the ball center) in the image center. We have not encountered convergence problems despite the limitations of the control of the joint's servomotors, the image segmentation errors and the linear approximation. The system is able to follow sequences of ball positions that involve recovering from "uncomfortable" configurations.

The structure of the paper is as follows. The notation used through the article is presented in Table 1. In Sect. 2 we classify our Visual Servoing according to the literature and provide a review of the basic control scheme. In Sect. 3 we provide a general description of the forward kinematics model. Section 4 gives the relevant geometrical description of the robot. In Sect. 5 we detail the construction of the Jacobian matrix that describes its forward kinematics model. Section 6 specifies the inversion of the forward kinematics model taking into account relevant constraints. In Sect. 7 we present empirical results on the actual performance of the approach. Section 8 gives our conclusions.

2 Background on Visual Servoing

There are two kinds of basic configurations for the camera [21], *eye-in-hand* and *fixed camera*. The eye-in-hand systems have the camera is placed on the final robot's effector, allowing to perform the video acquisition inside the working space,[1] centering the image capture process on the target object. The relation between the camera pose[2] and the robot's final effector pose is known and constant, because both share the same motion vector. The main disadvantage of this model is the possibility of loosing track of the target object as the robot movements may displace the target object outside the camera field of view. For this system, calibration of the camera intrinsic parameters may be required. The system considered in this paper is of this kind, where the robot's head is the system's effector. Figure 2 provides a diagram of the system.

[1] Working space is a region of task space where the robot's effector interacts with the objects it is working with.

[2] The pose is the physical position and orientation.

From a different point of view [32] Visual Servoing systems are classified into indirect and direct approaches. In indirect Visual Servoing systems there is a controller at each joint of the robot that performs the joint local control based on encoder information to reach the goal positions set by the Visual Servoing loop of the system. In direct Visual Servoing systems the robot joint positions are computed from the visual information at the sampling frequency of the camera. In this case the visual control loop includes the command of the joint servomotors. The system considered in this paper is of the *dynamic look-then-move* type. The joint and visual control loops are interleaved because the visual control loop allows to update the desired positions of the joints while the robot still continues executing the previous movement.

Finally, the literature distinguishes between Position Based Visual Servoing (PBVS) and Image Based Visual Servoing (IBVS). In the latter systems the values of the control parameters are computed as a direct function of the image features. For this reason, the IBVS is also known as 2D Visual Servoing system. The most common approach to bring the image feature parameter values to the desired ones is to perform a local minimization of the error based on a local linearization of the robot kinematic function. In this case there is a direct influence of . Therefore, the relationship between the image features on the state of the robot joints is specified through the image Jacobian matrix. In general, *a priori* knowledge about the image geometric features is needed, such as corners or edges [5, 8, 15] or visual landmarks [12].

2.1 General Visual Servoing Control Paradigm

An *image feature* is defined as any structural information that can be extracted from the image. Every feature corresponds to the projection of a real physical feature on the camera plane. From a set of k image feature parameters the *image feature parameter vector* is defined as $\mathbf{s} = [s_1, \ldots, s_k]^T$, with $\mathbf{s} \in \mathbb{R}^{\mathrm{k}}$.

2.1.1 Positioning Kinematic Error

The general robot control problem can be stated as reducing to zero of the positioning error $\|\mathbf{s}(\boldsymbol{\theta}, t) - \mathbf{s}^*\|$ in a finite time [27, 31]. The *Kinematic error function*, $e : \mathbb{R}^m \to \mathbb{R}^n$, is defined as:

$$e(\boldsymbol{\theta}, t) = \mathbf{C} \cdot (\mathbf{s}(\boldsymbol{\theta}, t) - \mathbf{s}^*), \tag{1}$$

where $\boldsymbol{\theta} \in \boldsymbol{T}$ is the vector of the robot joint positions, $\mathbf{s}(\boldsymbol{\theta}, t)$ is the vector of visual features, $\mathbf{s}^*$ the desired visual configuration, $\mathbf{C}$ the *combination matrix* of dimension $n \times m$, n is the number of degrees of freedom and m is the dimension of the state vector.

The *Task Jacobian* $J_e = \frac{\partial e}{\partial \boldsymbol{\theta}}$ is usually decomposed into $J_e = L_e \cdot J_r$ where $L_e = \frac{\partial e}{\partial \mathbf{r}} \in \mathbb{R}^{n\times 6}$ is the *Error Interaction matrix* relating the kinematic error with the robot end effector velocity, $J_r = \frac{\partial \mathbf{r}}{\partial \theta} \in \mathbb{R}^{6\times n}$ is the *Robot Jacobian* relating the final effector velocity with the velocities at the joints of the robot.

It is said that the task is *admissible* if there is an unique trajectory $\{\boldsymbol{\theta}(t), t \in [0, t^*]\}$ driving the error function to zero at the time limit ($e(\boldsymbol{\theta}, t^*) = \mathbf{0}$) and J_e is regular all over this trajectory. For the IBVS systems, admissibility requires the visibility condition, i. e. that there are enough visual features inside the vision range of the camera at any time.

2.1.2 Final Effector Trajectory in Task Space

It is assumed that the vector of visual features $\mathbf{s}$ is differentiable as a function of the robot's final effector pose $\mathbf{r}$. The velocity of the state vector can be expressed as a function of the velocity of the robot's final effector pose relative to the target object:

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial t} + \frac{\partial \mathbf{s}}{\partial t}. \tag{2}$$

Defining the *Feature Interaction matrix* $L_s = \frac{\partial \mathbf{s}}{\partial \mathbf{r}}$ as the Jacobian matrix that relates the visual features' velocity with the velocity of the robot's final effector in camera space, and $\mathbf{v}_r = \frac{\partial \mathbf{r}}{\partial t}$, (2) can be rewritten as:

$$\dot{\mathbf{s}} = L_s \cdot \mathbf{v}_r + \frac{\partial \mathbf{s}}{\partial t}. \tag{3}$$

Assuming that the combination matrix $\mathbf{C}$ does not depend on $\mathbf{r}$, the error interaction matrix can be rewritten as:

$$L_e = \mathbf{C} L_s. \tag{4}$$

An exponential decrease of the kinematic error (1) with time constant λ can be obtained applying the following linear differential equation:

$$\dot{e} = -\lambda e. \tag{5}$$

The time derivate of the kinematic error function, $\dot{e}$, can be written as a function of the velocity of the robot's joints, $\dot{\theta}$, and, if we use the decomposition $\frac{\partial e}{\partial \boldsymbol{\theta}} = \frac{\partial e}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \boldsymbol{\theta}}$, we get the following expression:

$$\dot{e} = \frac{\partial e}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \boldsymbol{\theta}} \dot{\boldsymbol{\theta}} + \frac{\partial e}{\partial t}, \tag{6}$$

which can be rewritten as:

$$\dot{e} = L_e . \mathbf{v}_r + \frac{\partial e}{\partial t}. \tag{7}$$

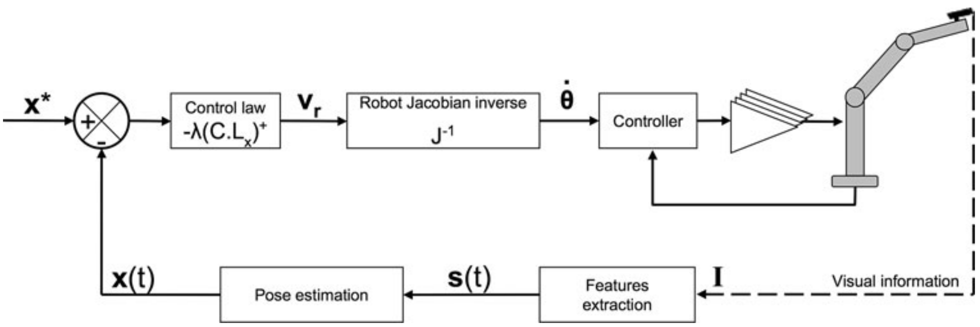


Fig. 2 General schema for Visual Servoing control

Isolating $\mathbf{v}_r$ from (7) we obtain the following expression:

$$\mathbf{v}_r = L_e^+\left(\dot{e} - \frac{\partial e}{\partial t}\right), \tag{8}$$

where L_e^+ is the pseudo inverse of the error interaction matrix.

Substitution of $\dot{e}$ in (8) by its expression in (5) lead us to the following expression for the robot's final effector velocity as a function of the kinematic error:

$$\mathbf{v}_r = -L_e^+\left(\lambda e + \frac{\partial e}{\partial t}\right). \tag{9}$$

However L_e and $\frac{\partial e}{\partial t}$ can only be estimated from the visual information, so the Cartesian velocity of the robot's final effector is formally defined as:

$$\widehat{\mathbf{v}_r} = -\widehat{L_e}^+\left(\lambda e + \widehat{\frac{\partial e}{\partial t}}\right). \tag{10}$$

2.1.3 *Stability and Convergence*

Substituting $\widehat{\mathbf{v}_\mathbf{r}}$ by (10) in (7), we get the following expression for the velocity of the kinematic error function:

$$\dot{e} = -\widehat{L_e}\widehat{L_e}^+\left(\lambda e + \widehat{\frac{\partial e}{\partial t}}\right) + \frac{\partial e}{\partial t}. \tag{11}$$

The sufficient condition assuring the steady decrease of the norm of the kinematic error function $\|e\|$ is $\widehat{L_e}\widehat{L_e}^+ > 0$. Generally the interaction matrix is chosen equal to the identity matrix, so the stability condition reduces to:

$$\widehat{L_s}\widehat{L_s}^+ > 0. \tag{12}$$

However, it must be taken into account that this convergence condition can be compromised by the inaccuracy of the robot effectors and the segmentation processes in the computer vision software, among other noise sources.

3 General Description of the Kinematics Model

We build a locally linear forward kinematics model of the robot by composing the diverse Jacobian matrices that embody the dependences among visual features and control parameters. Then we compute a simple inversion of the model to obtain the desired control commands that will accomplish the minimization of the visual error. In order to define the kinematics model a ground reference system is needed to formalize the effect of the joint motion on the visual features. The ground reference system is trivial for static manipulator robots, but it can be arbitrary and variable for legged robots. We have used the tips of the legs that are the actual ground contact points to define this ground reference system. We determine the ground plane using the robot joint's state information provided by the robot's basic control systems.

In Fig. 3 we show the tree structure of the decomposition of the Jacobian matrices which compose the linearized model of the system kinematics of a legged robot, following a hierarchical decomposition [14]. Each partial Jacobian matrix is a local linear system relating some input and output variables. The control commands are computed by inversion of the full Jacobian matrix.

The full system kinematics is described by the Jacobian matrix $J_{s\theta}$ which embodies the dependence of the visual features on the robot joints, it is decomposed as $J_{s\theta} = J_{sp}J_{p\theta}$. The Jacobian matrix J_{sp} embodies the dependence of the image features on the upper body articulations and the extended support points. The Jacobian matrix $J_{p\theta}$ embodies the dependence of the extended support point positions and upper body articulations on the robot articulations.

The Jacobian matrix J_{sp} is further decomposed as $J_{sp} = J_{sc}J_{cp}$, where J_{sc} embodies the dependence of the image features on the camera reference system and J_{cp} embodies the dependence of the camera reference system on the upper body degrees of freedom and the extended support points.

The Jacobian matrix J_{sc} is constructed by aggregating the Jacobian matrices corresponding to each feature point into a block diagonal matrix. The Jacobian matrix J_{cp} is also a diagonal aggregation of matrices $J_{c\theta_u}$, that embodies the dependence of the camera reference system on the upper articulations, and J_{ce}, that embodies the dependence of the camera reference system on the extended support point positions.

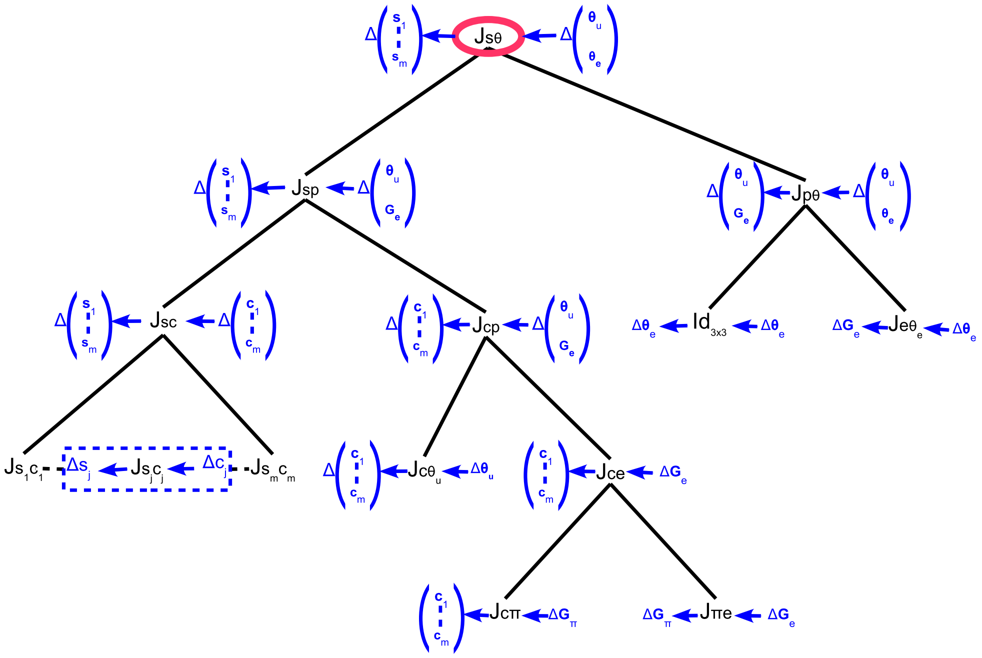


Fig. 3 General structure of the Jacobian operators composing the direct kinematics model

The Jacobian matrix J_{ce} is further decomposed as $J_{ce} = J_{c\pi} J_{\pi e}$, where $J_{c\pi}$ embodies the dependence of the camera reference system on the basic support point positions, and $J_{\pi e}$ embodies the dependence of the basic support point positions on the extended support point positions.

Finally, the Jacobian $J_{p\theta}$ is a diagonal aggregation of the identity matrix $\mathbf{I}_{3\times 3}$ and the Jacobian matrix $J_{e\theta}$, that embodies the dependence of the extended support point positions on the robot articulations.

4 Robot's Geometry

Specifying the robot's geometry is required to formulate the robot direct kinematics. First we describe how we compute the support points, then we describe the coordinate reference systems and the transformations between them. Finally, we present the image feature coordinates

4.1 Support Points

The legs define the relationship between the robot's body and the supporting plane (Fig. 4) where it is standing on. The *support points* denoted $\mathbf{g}_i$, are the leg points in contact with the ground. They are highlighted by red dots in Fig. 4. Each leg has, at most, a support point, which can be the tip or one of the knees. When the robot is standing at least three legs must have their supporting points in contact with the ground. The supporting points 3D coordinates must be computed relative to the robot body center. In order to determine which leg tips are the supporting points, we proceed as follows: (1) We compute the tip position of each leg relative to the body center coordinate reference system from the knowledge of the joint angles of each leg articulation. (2) We compute the hypothetical supporting planes defined by each combination of three leg tip points. (3) We discard hypothetical supporting planes for which at least one leg tip is *below* it or do not comply with the stability condition about the center of mass.

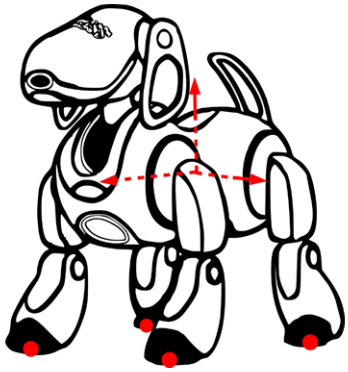


Fig. 4 Support Points and body coordinate reference system

First we compute the position of the leg tip $\mathbf{g}$ applying the transformations defined by the chain of articulations from the leg tip up to the body center, as shown in general in Fig. 5. Figure 6 shows the leg parameter specification for the Aibo robot. The In homogeneous coordinates, the leg tip is computed by the following product of elemental transfor-

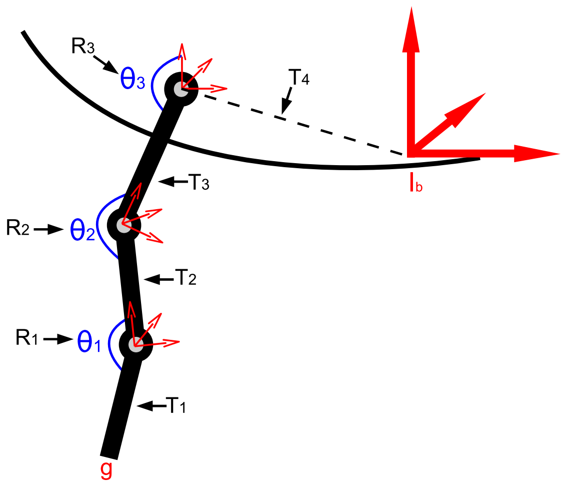


Fig. 5 Leg geometry. Transformations applied to determine the leg tip coordinates

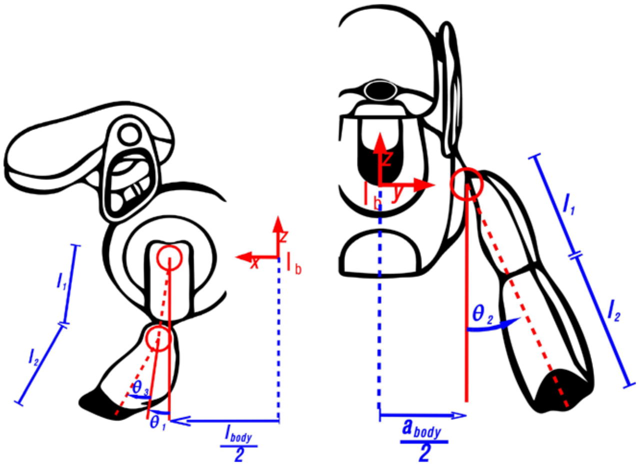


Fig. 6 Geometry of the leg's articulations in the Aibo robot. Lateral and frontal views

mation matrices:

$$\begin{pmatrix}\mathbf{g}\\ 1\end{pmatrix} = (\mathbf{T}_{n+1}\cdot \mathbf{R}_n.\mathbf{T}_n \cdots \mathbf{R}_1.\mathbf{T}_1).\begin{pmatrix}\mathbf{0}\\ 1\end{pmatrix}, \qquad (13)$$

where $\mathbf{R}_k$ is the rotation matrix corresponding to the k-th leg articulation from tip to the body center, being n the number of articulations and $\mathbf{T}_{k-1}$ the translation matrix corresponding to the leg segment between the $(k-1)$-th and the k-th articulations. Translation matrix $\mathbf{T}_1$ corresponds to the translation from the tip to the first articulation, while translation matrix $\mathbf{T}_{n+1}$ corresponds to the translation from the last articulation to the body center reference system. For a given robot's leg, we denote $\mathbf{L} = (\mathbf{T}_{n+1}\cdot \mathbf{R}_n.\mathbf{T}_n \cdots \mathbf{R}_1.\mathbf{T}_1)$. the transformation giving the leg's tip.

Each subset of three leg tip points $\mathbf{G}_\pi = (\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3)$ allows to compute the parameters (a, b, c, d) of a hypothetical support plane solving equation $\pi(\mathbf{g}_i) = a\mathbf{g}_{i,x} + b\mathbf{g}_{,iy} + c\mathbf{g}_{i,z} + d = 0$. If all of the remaining leg tip points $\mathbf{g}'$ we have $\pi(\mathbf{g}') > 0$ then it is a support plane corresponding to the true ground surface. In order for the robot to be standing in a stable pose, the projection of the body mass center, according to the direction of gravity, must lie inside of the triangle defined by the support points [2]. This condition is illustrated in Fig. 7. Assuming a set of *basic support points* $\mathbf{G}_\pi$ the set of *extended support points* is defined as: $\mathbf{G}_e = \{\mathbf{g}\ s.t.\ |\pi(\mathbf{g})| < tol\}$, where tol is a tolerance limit for the distance between the ground plane and a leg tip point in order to accept it as a support point.

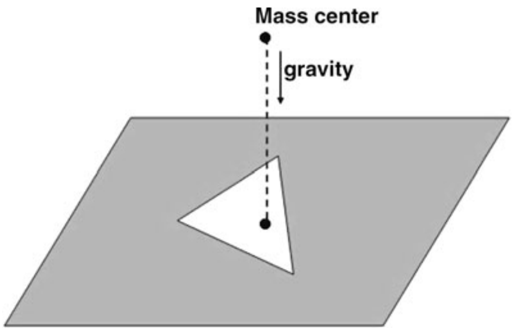


Fig. 7 Stability Condition to determine the basic support points on the ground plane

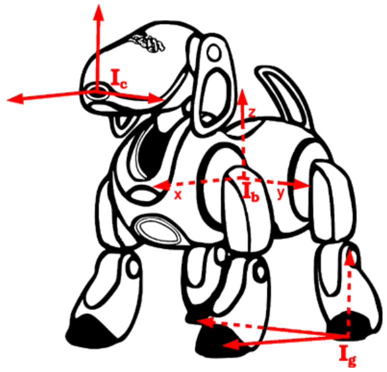


Fig. 8 Coordinate reference systems I_g, I_b, I_c on the Aibo robot

4.2 Coordinate Reference Systems

In order to propagate the correction computed to minimize the visual feature error, it is necessary to define the relevant coordinate reference systems coexisting in the robot, and the transformations between them. We denote I_a a generic reference system, and ${}_aI_b$ the transformation from I_b to I_a. The three coordinate reference systems of interest in our application, illustrated in Fig. 8 for the Aibo robot, are:

- The fixed reference system whose origin lies on the ground, I_g. Previous works in the literature [19] assume a predefined world reference system. To avoid this limitation, we define this system relative to the basic ground support points defined in Sect. 4.1.

- The body coordinate reference system, I_b, whose origin is the geometrical center of the robot's body. It is assumed that all the readings from the system configuration (i.e. leg configurations) are provided in this frame of reference.
- The camera reference system, I_c.

4.2.1 *The Ground Reference System I_g*

We build, in the reference system I_b, the expression of the director vectors which define the axes of I_g. Assuming a set of basic support points positions $\mathbf{G}_\pi$, we arbitrarily define $\mathbf{g}_1$ as the origin of I_g. The vectors $\overrightarrow{\mathbf{g}_1\mathbf{g}_2}$ and $\overrightarrow{\mathbf{g}_1\mathbf{g}_3}$ define the direction of the two first reference axes of $I_g$lying on the ground supporting plane, note that they may not be orthogonal. We built the third axis director vector as their cross product, orthogonal to the ground supporting plane.

4.2.2 *Transformation ${}_bI_g$ Between I_g and I_b*

The first component of the transformation ${}_bI_g$ is the rotation matrix $\mathbf{R}_0$, built from the three director vectors defining the axes of I_g:

$$R_0 = \begin{pmatrix} \mathbf{g}_2 - \mathbf{g}_1 & \mathbf{g}_3 - \mathbf{g}_1 & (\mathbf{g}_3 - \mathbf{g}_1) \times (\mathbf{g}_2 - \mathbf{g}_1) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{14}$$

and the second is the translation from the origin of I_b to the origin of I_g,

$$T_0 = \begin{pmatrix} I_{3\times 3} & \mathbf{g}_1 \\ 0 & 1 \end{pmatrix}. \tag{15}$$

So, composing the two transformations we finally obtain the matrix transformation from I_g to I_b,

$${}_bI_g = \mathbf{T}_0\mathbf{R}_0. \tag{16}$$

4.2.3 *Transformation ${}_cI_b$ Between I_b and I_c*

Figure 9 shows the neck and head geometric parameters. There are two tilting degrees of freedom of the Aibo, denoted θ_{tilt} and θ_{nod}. The first corresponds to the neck base pivoting over the chest, the second allows corresponds to the head moving vertically at the head-neck articulation. The third degree of freedom, called θ_{pan}, allows a rotation perpendicular to the tilt, moving the head from side to side. The transformation between the camera and body reference systems is the following composition:

$${}_bI_c = T_3R_2T_2R_1T_1, \tag{17}$$

where T_1 is the translation from the camera base to the top of the neck, R_1 accounts for the *nod* and *pan* head rotations, T_2 is the translation from the neck top joint to the neck base joint, R_2 accounts for the neck-head tilt rotation, and T_3 is the translation from the neck base to the body mass center.

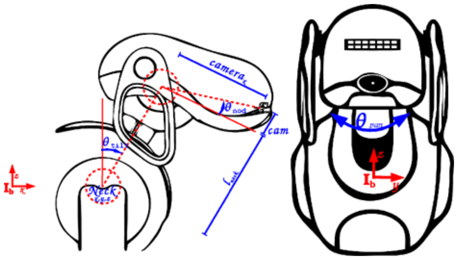


Fig. 9 Aibo head's degrees of freedom. Lateral and front views

4.3 Image Features

The image plane coordinates of the visual feature $\mathbf{s}_i = (u_i, v_i)^T$ are computed from the feature coordinates ${}^c\mathbf{s}_i = (x_i, y_i, z_i)^T$ in the camera system I_c, according to the projective equation:

$$s_i = \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \frac{\lambda}{x_i} \begin{pmatrix} y_i \\ z_i \end{pmatrix} = \phi({}^c\mathbf{s}_i), \tag{18}$$

where λ is the focal length. We assume that the featured object is static relative to the ground reference system I_g, therefore we can obtain the image plane coordinates $\mathbf{s}_i$ from the corresponding coordinates ${}^g\mathbf{s}_i$ in I_g applying the transformation ${}_cI_g$ between I_g and I_c, defined as the composition of transformations ${}_cI_b$ and ${}_bI_g$.

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \phi({}_cI_g({}^g\mathbf{s}_i)). \tag{19}$$

4.3.1 *Aibo's Image Feature Vector*

In the experiments of Sect. 7 the stated task goal is to bring the ball to the image center, therefore the image feature $\mathbf{s} = (u, v)$ is the center of the region identified as the ball by the Aibo's segmentation algorithms. We also consider the relation between the known ball diameter D and its estimation from the segmentation of the image $\hat{D}$, to compute an estimation $\hat{d}$ of the ball distance to the camera:

$$\hat{d} = \lambda \frac{1}{2} \frac{D}{\tan(\frac{1}{2}\hat{D})}. \tag{20}$$

The estimated distance is used to compute the feature camera coordinates from the image feature coordinates:

$${}^c\mathbf{s} = \hat{d} \begin{pmatrix} \cos(v).\cos(u) \\ \sin(u) \\ \sin(v) \end{pmatrix}. \tag{21}$$

5 Direct Kinematics

We construct the Jacobian matrix $J_{s\theta}$ relating the variations of the robot degrees of freedom with the variations of the visual features in the image plane composing partial influences among the robot components and with the visual features according to the general strategy described in Sect. 3 and illustrated in Fig. 3.

5.1 Dependence of Image Features on the Camera J_{sc}

Deriving (18) we get the following differential equation for an image feature $\mathbf{s}_i$:

$$\begin{pmatrix} \dot{u}_i \\ \dot{v}_i \end{pmatrix} = \begin{pmatrix} \frac{-\lambda . y_i}{x_i^2} & \frac{\lambda}{x_i} & 0 & 0 \\ \frac{-\lambda . z_i}{x_i^2} & 0 & \frac{\lambda}{x_i} & 0 \end{pmatrix} \cdot \begin{pmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \\ 0 \end{pmatrix} = J_{s_i c}\, {}^c\dot{\mathbf{s}}_i . \tag{22}$$

To build the Jacobian matrix J_{sc} that relates the variations of the features in camera space with their image projections, we aggregate the individual feature Jacobian matrices $J_{s_i c}$ into the following block diagonal matrix:

$$J_{sc} = \begin{pmatrix} J_{s_1 c} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & J_{s_k c} \end{pmatrix} . \tag{23}$$

Therefore, J_{sc} defines a linear transformation from variations of feature point positions in the camera reference system I_c into variations of the image feature vector $\mathbf{s}$,

$$\Delta \mathbf{s} \simeq J_{sc} \cdot \Delta({}^c\mathbf{s}).$$

5.2 Dependence of the Basic Support Points $\mathbf{G}_\pi$ on the Extended Support Points $\mathbf{G}_e$: $J_{\pi e}$

Changes in the extended support points $\triangle \mathbf{G}_e$ may induce changing the ground plane and changes $\triangle \mathbf{G}_\pi$ in the basic support points:

$$\triangle \mathbf{G}_\pi = J_{\pi e} \triangle \mathbf{G}_e ,$$

which can be expanded as follows:

$$\begin{pmatrix} \triangle \mathbf{g}_1^\pi \\ \triangle \mathbf{g}_2^\pi \\ \triangle \mathbf{g}_3^\pi \end{pmatrix} = \begin{pmatrix} M_{11} & \cdots & M_{1n} \\ M_{21} & \cdots & M_{2n} \\ M_{31} & \cdots & M_{3n} \end{pmatrix} \begin{pmatrix} \triangle \mathbf{g}_1^e \\ \vdots \\ \triangle \mathbf{g}_n^e \end{pmatrix} , \tag{24}$$

where variations on the ground plane are expanded as $\triangle \mathbf{G}_\pi = (\triangle \mathbf{g}_1^\pi, \triangle \mathbf{g}_2^\pi, \triangle \mathbf{g}_3^\pi)^T$, variations on the extended support points are expanded as $\triangle \mathbf{G}_e = (\triangle \mathbf{g}_1^e, \ldots, \triangle \mathbf{g}_n^e)^T$ and $J_{\pi e} \in \mathbb{R}^{12 \times 4n}$ is the Jacobian matrix of (24). The Jacobian building block matrices of size 4 × 4 are defined as follows:

- $M_{ij} = \mathbf{I}_{4\times 4}$, if the tip of the j-th leg corresponds to the i-th basic support point.
- $M_{ij} = \mathbf{0}$, if the tip of the j-th leg does not correspond to the i-th basic support point.

5.3 Dependence of Camera Coordinates on Upper Body Joint Angles and Extended Support Points J_{cp}

According to (19), the image feature coordinates in the camera reference system could be expressed as a function of the feature coordinates in the ground reference system. Deriving (19) we get the Jacobian matrix that relates the variations in the camera reference system I_c with the variations in the upper body articulations and the basic support points positions:

$$J_{cp} = \frac{\partial({}^c\mathbf{s})}{\partial \mathbf{p}_\pi} = \frac{\partial({}_c I_b \cdot {}_b I_g({}^g\mathbf{s}))}{\partial \mathbf{p}_\pi} , \tag{25}$$

where $\mathbf{p}_\pi$ is the following vector of upper joints and basic support points positions:

$$\mathbf{p}_\pi = \begin{pmatrix} \boldsymbol{\theta}_u \\ \mathbf{G}_\pi \end{pmatrix} . \tag{26}$$

Using the chain rule, we rewrite (25):

$$J_{cp} = \frac{\partial({}_c I_b)}{\partial \mathbf{p}_\pi} ({}_b I_g)({}^g\mathbf{s}) + ({}_c I_b) \frac{\partial({}_b I_g)}{\partial \mathbf{p}_\pi} ({}^g\mathbf{s}) . \tag{27}$$

The transformation ${}_c I_b$ depends only on $\boldsymbol{\theta}_u$ (the upper body joint angles) and ${}_b I_g$ depends on $\mathbf{G}_\pi$. We can define the following independent Jacobians from (27):

$$J_{c\theta_u} = \frac{\partial({}_c I_b)}{\partial \boldsymbol{\theta}_u} ({}_b I_g)({}^g\mathbf{s}) , \tag{28}$$

$$J_{c\pi} = ({}_c I_b) \frac{\partial({}_b I_g)}{\partial \mathbf{G}_\pi} ({}^g\mathbf{s}) , \tag{29}$$

where $J_{c\theta_u}$ defines the dependences of the features in the camera reference system on the upper body articulations and $J_{c\pi}$ defines the dependences of the features in the camera reference system on the basic support points. These matrices corresponding to the kinematics in two orthogonal subspaces.

If we make use of matrix $J_{e\pi}$, defined in (24), we can define the dependences of the camera on the extended support points:

$$J_{ce} = J_{c\pi} \cdot J_{\pi e} . \tag{30}$$

The Jacobian matrix J_{cp} can be constructed as the aggregation of the Jacobian matrices $J_{c\theta_u}$ and J_{ce} into the following block diagonal matrix:

$$J_{cp} = \begin{pmatrix} J_{c\theta_u} & 0 \\ 0 & J_{ce} \end{pmatrix} . \tag{31}$$

The dependence of the variations in the camera feature positions on the variations in the upper body degrees of freedom and the extended support points positions can be summarized by:

$$\Delta({}^{c}\mathbf{s}) \simeq J_{cp}\Delta\mathbf{p}, \tag{32}$$

where $\mathbf{p}$ is the vector of upper joint angles and extended support point positions:

$$\mathbf{p} = \begin{pmatrix} \theta_u \\ \mathbf{G}_e \end{pmatrix}. \tag{33}$$

5.4 Dependence of $\mathbf{p}$ on the Robot's Articulation Angles: $J_{p\theta}$

We can decompose the Jacobian matrix $J_{p\theta}$, which defines the dependence of $\mathbf{p}$ on the robot's degrees of freedom, into two Jacobian matrices, one which relates the extended support points and the leg's articulations $J_{e\theta}$ and the other which is the identity matrix for the upper body articulations. To build $J_{e\theta}$ we start modelling the changes in the extended support point coordinates in response to the changes in the degrees of freedom of the corresponding leg, as follows:

$$\Delta\mathbf{g}_i^e \simeq J_i \cdot \Delta\boldsymbol{\theta}_i, \tag{34}$$

where J_i is the Jacobian matrix of equation (13), relating the variations of the i-th extended support point to variations of the leg's joint angles $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2}, \ldots, \theta_{im_i})$ in the leg corresponding to this i-th extended support point. The size of each J_i matrix is $4 \times m_i$, being m_i the number of joints of leg i.

Aggregating the extended support point Jacobians into a diagonal block matrix, we get the following equation:

$$\begin{pmatrix} \triangle\mathbf{g}_1^e \\ \vdots \\ \triangle\mathbf{g}_n^e \end{pmatrix} = \begin{pmatrix} J_1 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & J_n \end{pmatrix} \begin{pmatrix} \triangle\boldsymbol{\theta}_1 \\ \triangle\boldsymbol{\theta}_2 \\ \vdots \\ \triangle\boldsymbol{\theta}_n \end{pmatrix}, \tag{35}$$

which can be expressed in matrix form as follows:

$$\triangle\mathbf{G}_e = J_{e\theta}\Delta\boldsymbol{\theta}_e, \tag{36}$$

where $\boldsymbol{\theta_e}$ is the vector composed of all the joint angles of the legs corresponding to all the extended support points.

In order to obtain a single matrix that relates the variations of $\mathbf{p}$ in all the articulation joints of the robot, we define the following diagonal block matrix:

$$J_{p\theta} = \begin{pmatrix} \mathbf{I}_{m\times m} & 0 \\ 0 & J_{e\theta} \end{pmatrix}, \tag{37}$$

where m is the number of upper body articulations.

The dependence of variations in vector $\mathbf{p}$ on variations in the robot degrees of freedom, can be summarized as:

$$\Delta\mathbf{p} \simeq J_{p\theta}\Delta\boldsymbol{\theta}. \tag{38}$$

We must note that for the Aibo, we can have as supporting points the knees. Then the Jacobians in (34) may have two different expressions. Considering this issue in detail will introduce irrelevant complexity.

5.5 Image Feature Jacobian Matrix

Finally, we construct the full Jacobian matrix that models the dependence of the image features on all the degrees of freedom of the robot by composing the Jacobian matrices of (23), (31) and (38):

$$\Delta\mathbf{s} = (J_{sp}J_{p\theta})\Delta\theta, \tag{39}$$

where $J_{sp} = J_{sc}J_{cp}$. The global Jacobian matrix is defined as:

$$J_{s\theta} = J_{sp}J_{p\theta}. \tag{40}$$

Equation (39) is thus rewritten in the following way:

$$\Delta s = J_{s\theta}\Delta\theta. \tag{41}$$

6 Constrained Inverse Kinematics

To determine the velocity at each robot's degrees of freedom that will minimize the dynamic visual error $\|\dot{s} - (J_{s\theta})\dot{\theta}\|$, we must compute the inverse of the $J_{s\theta}$ matrix in (41). In general, this matrix is not invertible, because it is underconstrained. The general minimum least squares solution is given by the seudoinverse $J_{s\theta}^{+}$ in the following way:

$$\dot{\boldsymbol{\theta}} = J_{s\theta}^{+}\dot{\mathbf{s}} + (I - J_{s\theta}^{+}J_{s\theta})\mathbf{w}, \tag{42}$$

where $\mathbf{w}$ is an arbitrary vector of R^{m+3n}, m is the number of upper body joints and n is the number of extended support points. In general, $(I - J_{s\theta}^{+}J_{s\theta})\mathbf{w} \neq 0$ and all the vectors of the form $(I - J_{s\theta}^{+}J_{s\theta})\mathbf{w}$ belong to the kernel of the transformation associated to $J_{s\theta}$. However, (42) does not take into account the constraint of preserving the ground reference system invariant, which implies keeping the distances between supporting points constant.

We denote $\mathbf{d} = (d_1, \ldots, d_{n(n-1)})^T$ a vector containing all distances among the extended support points. The Jacobian matrix that relates the changes in this vector to the changes in the extended support points is given by:

$$J_{de} = \left(\frac{\partial d_i}{\partial \mathbf{e}_j}; i = 1, \ldots, n(n-1); j = 1, \ldots, n\right). \tag{43}$$

The vector $\mathbf{d}$ is not affected by variations in the upper body's degrees of freedom, thus we can formulate its dependency on $\mathbf{p}$ by a Jacobian matrix J_{dp} constructed as follows:

$$J_{dp} = \begin{pmatrix} 0_{n\times m} & 0 \\ 0 & J_{de} \end{pmatrix}. \tag{44}$$

$$\Delta\mathbf{d} \simeq J_{dp} \cdot \begin{pmatrix} \Delta\theta_u \\ \Delta\mathbf{G_e} \end{pmatrix}. \tag{45}$$

Using J_{dp}, J_{pc} and J_{sc} we can construct the dependencies of $\mathbf{d}$ on the variations of image features:

$$\Delta\mathbf{d} = [J_{dp} J_{pc} J_{sc}^{+}]\Delta\mathbf{s}.$$

We want to our control system to preserve $\mathbf{d}$. Therefore, we must project the obtained movements into the null space of $[J_{dp} J_{pc} J_{sc}^{+}]$, to this end we define the constrained control rule for the movements of the extended support points:

$$\Delta\mathbf{p}^1 = [(I - J_{dp}^{+} J_{dp})(J_{pc} J_{sc}^{+})] k_s \Delta\mathbf{s}, \tag{46}$$

where $k_s \in \mathbb{R}$ is a control gain.

However, due to noise and the inaccuracy of the linear approximations and positioning systems, the actual motion of the robot articulation induce undesired variations in $\mathbf{d}$. Therefore, some corrective actions for repositioning the support points are required, we project the movements obtained by applying the seudoinverse of Jacobian J_{dp} to the error in the distances between extended support points positions into the null space of J_{sp}^{+}.:

$$\Delta\mathbf{p}^2 = [(I - J_{sp}^{+} J_{sp}) J_{dp}^{+}] k_d \Delta\mathbf{d}, \tag{47}$$

where $k_d \in \mathbb{R}$ is a control gain.

Finally, we combine (46) and (47) in order to obtain a control law that moves the articulations maintaining the ground reference system invariant while moving the image features to the desired ones. This global control law is defined as follows:

$$\Delta\theta = J_{p\theta}^{+}[\Delta\mathbf{p}^1 + \Delta\mathbf{p}^2]. \tag{48}$$

This equation allows us to determine the variations on the robot's degrees of freedom to get the desired configuration of the image. However, this equation is unrestricted and may drive the robot into unstable configurations, that is, to articulation configurations out of the region of stable poses in configuration space. Stable poses are characterized by the support points convex hull condition [2] illustrated in Fig. 7. When this condition does not hold or the projection point is too close to the polygon boundary we restrict the Visual Servoing to the upper body degrees of freedom, using the transformation ${}_g I_c$ instead of ${}_b I_c$ to construct a reduced Jacobian $J_{s\theta_u}$ that relates the image features to the upper body degrees of freedom [25, 30].

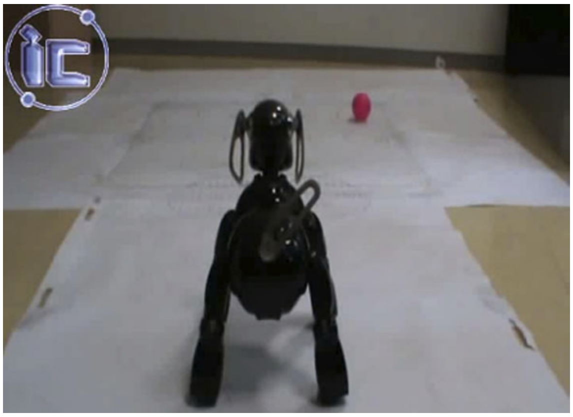


Fig. 10 The experimental ground

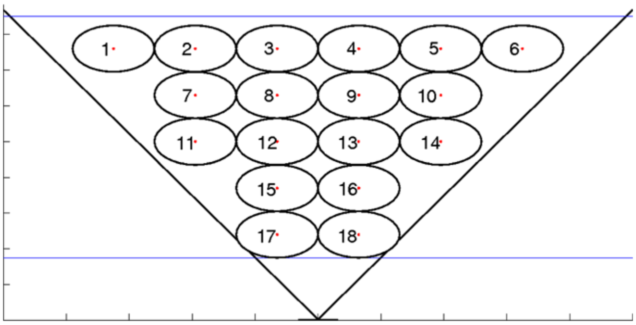


Fig. 11 Ground plane partition in front of the Aibo. Each ellipse models the uncertainty associated to its center location. Aibo's position is in the lower vertex of the triangle

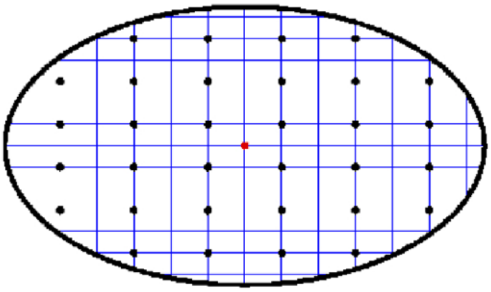


Fig. 12 Discretization of the space inside the uncertainty ellipse for each location

7 Empirical Results on the Aibo

We have applied the Visual Servoing scheme to the visual tracking of a ball using all the degrees of freedom of a Sony's Aibo ERS-7 robot as described in Sect. 4.3.1. The visual feature is the center of the image region labeled as the ball by the robot's computer vision algorithms. Program development has been done using the Carnegie Mellon University's SDK [34] and the SONY's SDK [7]. We examine the behavior of the robot under different controlled experimental settings. Figure 10 shows the experimental ground,

Table 2 Average final visual error at each uncertainty circle

Nominal	Error norm	
position	Average	Variance
1	0.0821	0.0024
2	0.0732	0.0007
3	0.0642	0.0011
4	0.0886	0.0097
5	0.0757	0.0006
6	0.0874	0.0062
7	0.0704	0.0008
8	0.0791	0.0016
9	0.0935	0.0016
10	0.0860	0.0002
11	0.0766	0.0010
12	0.0868	0.0056
13	0.0698	0.0016
14	0.0737	0.0008
15	0.0740	0.0097
16	0.0880	0.0004
17	0.0692	0.0038
18	0.0665	0.0006
Total	0.0783	0.0027

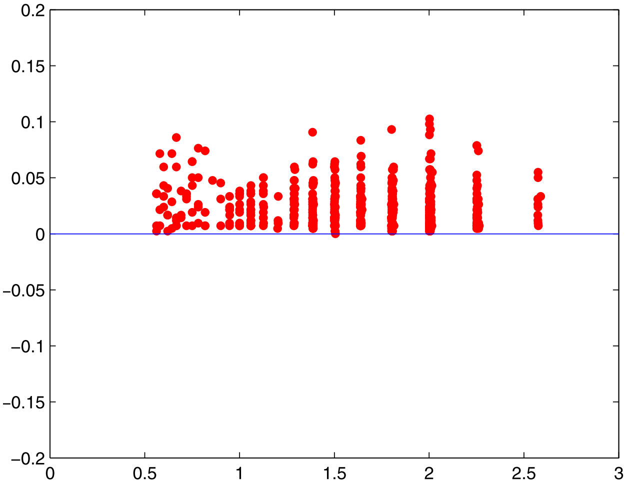


Fig. 13 Final Visual Servoing error norm distribution versus distance to the ball in the 3D world reference system

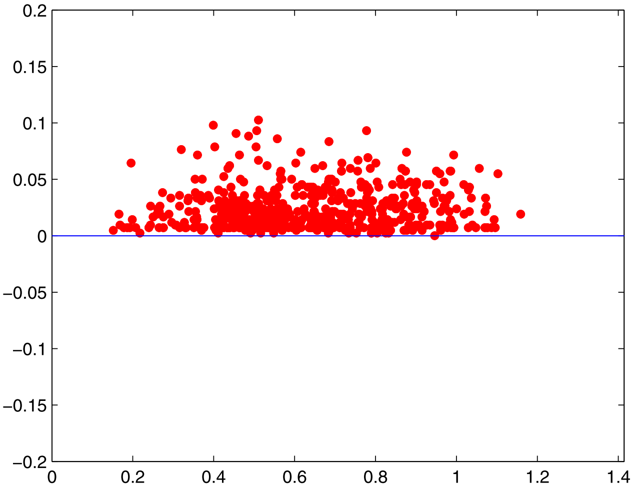


Fig. 14 Final visual error norm versus initial visual error

Table 3 Accumulated derivatives of the error trajectory at each nominal position

Nominal	Derivative Norm	
position	Average	Variance
1	0.0065	0.0016
2	0.0065	0.0003
3	0.0059	0.0002
4	0.0065	0.0009
5	0.0052	0.0005
6	0.0089	0.0027
7	0.0061	0.0004
8	0.0050	0.0002
9	0.0059	0.0003
10	0.0090	0.0006
11	0.0050	0.0005
12	0.0029	0.0002
13	0.0051	0.0007
14	0.0066	0.0005
15	0.0051	0.0002
16	0.0030	0.0002
17	0.0029	0.0002
18	0.0011	0.0000
Total	0.0052	0.00054

where the paper sheet contains the appropriate marks for the measured experimental locations. A sample video of the experiment is available.[3] The experiments start from nominal initial values of the Aibo's degrees of freedom in a standing stable position. The Aibo position and orientation defines the world reference system. The two experiments are as follows:

- Experiment 1: The ball is placed in a fixed position. The robot performs Visual Servoing to place the ball center in the image plane center, stopping after reaching the goal under some tolerance condition. The ground plane before the robot is partitioned as specified in Fig. 11 to allow for the systematic test of the robot's behavior sensitiv-

[3] http://www.ehu.es/ccwintco/index.php/DPI2006-15346-C03-03-Resultados#videos.

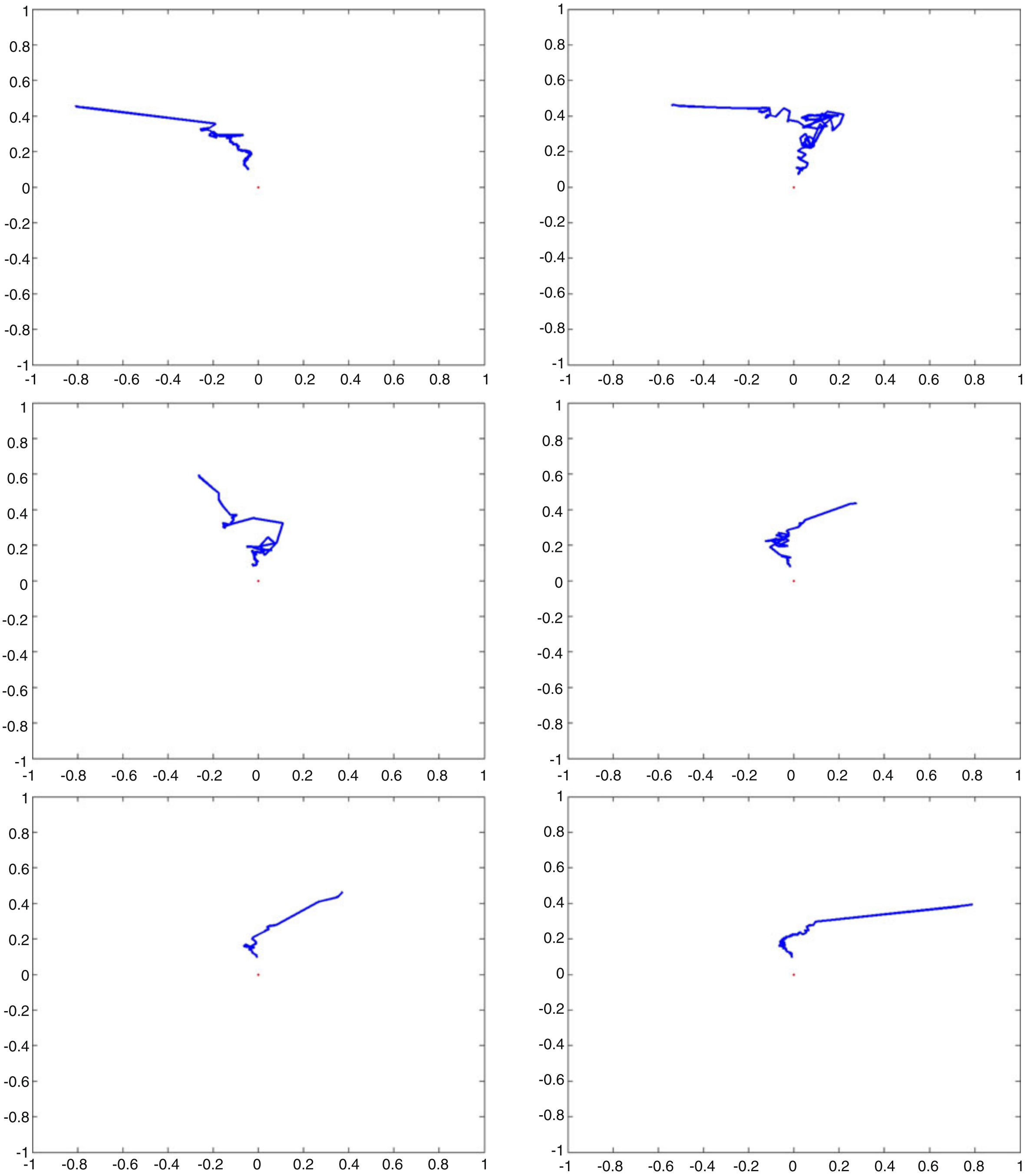


Fig. 15 Sample trajectories of the ball center in the image plane with the ball placed in some position inside the uncertainty circle (positions 1 to 6)

ity to the positions of the ball. Each location ellipse is further discretized to simulate uncertainties in ball position.

- Experiment 2: The ball is placed in a sequence of positions. The robot performs Visual Servoing at each position. After normal stopping the ball is moved to the next position in the sequence. The aim of this experiment is to test the ability of the Visual Servoing to move the robot smoothly from one final configuration of its articulations to another.

7.1 Experiment 1: Visual Tracking of a Static Ball

The reference distance between sampling points in the (x, y) plane is 30 cm. The horizontal angular aperture of the cam-

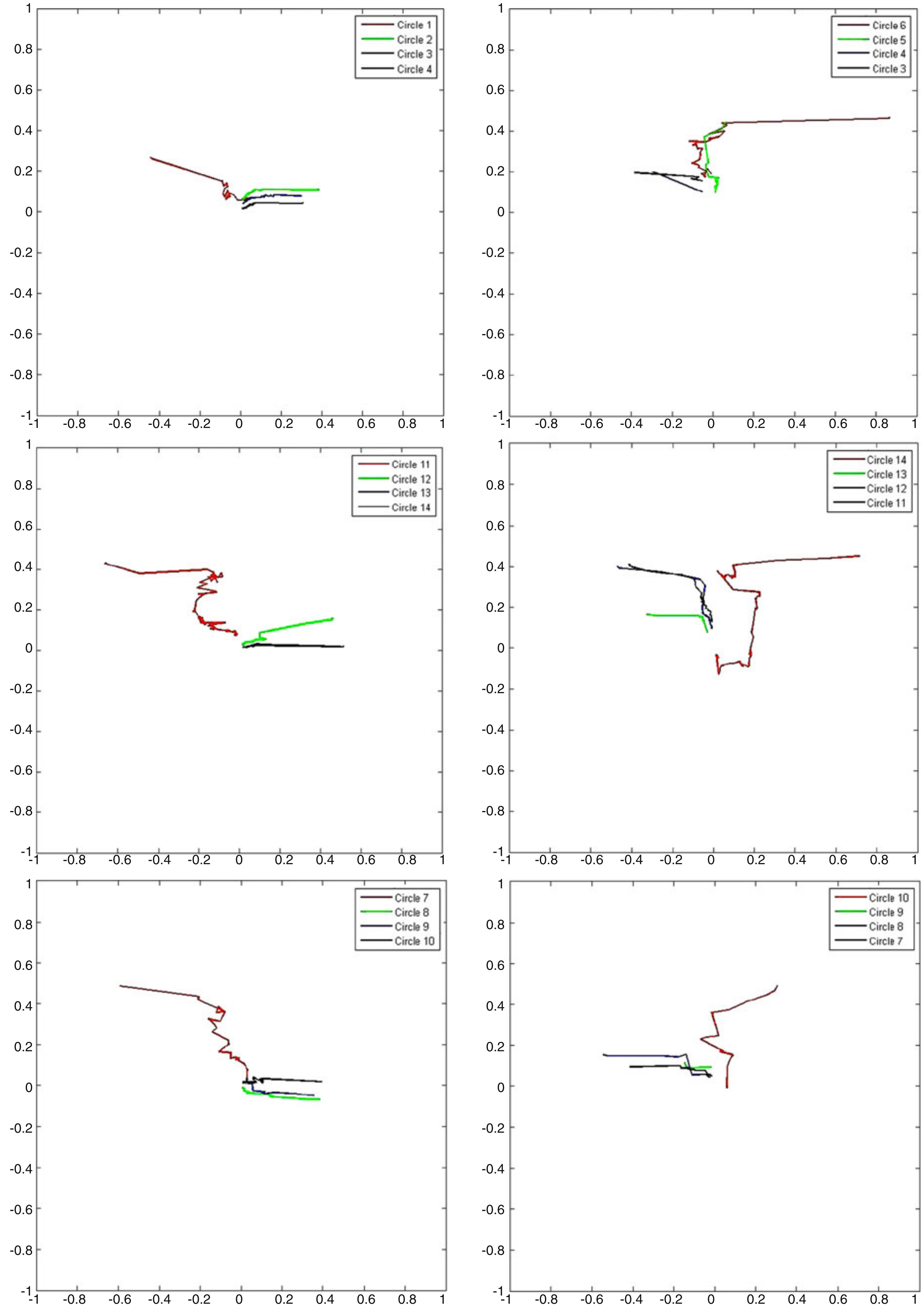


Fig. 16 Trajectories in the image plane when the visual servoing is restarted after moving the ball

era's field of view is $\frac{\pi}{3}$ radians. The distance to the camera is in the range from 0.5 m to 2 m. The projection on the ground of the Aibo's center of mass in its nominal initial configuration is assumed as the origin of the world reference system. The ground in front of the robot which is inside the robot's field of view is partitioned, as illustrated in Fig. 11, into the uncertainty ellipses of 18 nominal positions which are further discretized as shown in Fig. 12. Each point corresponds to a position where we place the ball simulating perturbations of the nominal position which is the ellipse center. We have a total of 576 points into the vision range of the Aibo robot where we can place the ball to perform the Visual Servoing experiments.

The range of values of the image plane coordinates is $[-1, 1]$ for both axes. After testing all the possible positions for the ball, the average norm of the final error in the image plane is 0.0783 with a variance of 0.0027, the resolution of the Visual Servoing is below 4% of the image space on average. Table 2 details the average and variance of the final visual error for each ball position.

Figure 13 plots the visual error norm at the end of the Visual Servoing process versus the distance from the ball to the camera image plane. The magnitude of the error does not show any trend related to the distance to the camera, the Visual Servoing performance is nearly invariant relative to the distance of the target object to the camera. The vertical structures in the plot correspond to a collection of experiments where the ball was perceived at similar distances. Those structures show the same uniform distribution of the visual error regardless of distance. Therefore, the distribution of the final error is invariant to the distance of the ball to the robot. In Fig. 14 we plot the norm of the final visual error after Visual Servoing versus the initial visual error. The figure does not show column structures as Fig. 13, because we have a nearly uniform distribution of the initial visual error. We find that the final visual error is invariant to the initial visual error. In Fig. 15 we show some sample trajectories of the ball center in the image plane along the Visual Servoing process. Most of them show a fairly smooth convergence to the image center

To assess the smoothness of the Visual Servoing trajectories we have computed the average spatial derivative of the trajectories, starting from the sampling points in each uncertainty circle associated with each ground plane sampling position. These values are shown in Table 3. We can appreciate that some of the starting ball positions produce systematically smooth trajectories, such as position 17 and 18, while others, such as positions 6 and 10, produce more jumpy trajectories. The reasons for that behaviour lie in the uneven distribution of the servomotors response power and control resolution on the Aibo's articulations, as well as the image segmentation problems.

7.1.1 *Visual Tracking for a Sequence of Ball Positions*

In the second experiment, we tested several sequences of positions of the ball, each one consisting of four positions. The Aibo performed the Visual Servoing starting from the robot configuration after Visual Servoing for the previous ball position, no corrections were made. The ball was static while the Aibo was performing each Visual Servoing process. The aim of this experiment is to study the degradation of performance due to the accumulation of errors. Figure 16 shows some example trajectories obtained in this experiment. In the plots, the red, green, blue and black trajectories correspond to the Visual Servoing trajectories performed by the Aibo after each of the four ball positions. It can be appreciated that the robot response is quite smooth for the ensuing positions after having performed the Visual Servoing for the first one, even if the initial position was a "difficult" one.

8 Conclusions

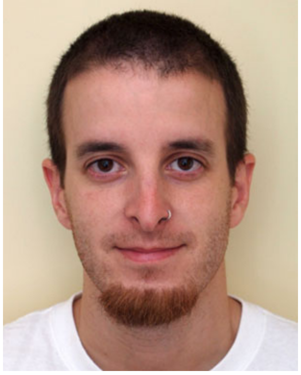
We have developed the Visual Servoing for the whole set of degrees of freedom of the Aibo ERS-7 following a principled approach. From the geometrical description of the robot we have constructed the full Jacobian matrix that embodies the functional dependence of the visual features on the robot joint angles. We take into account also the correction of the motion of the supporting points. The seudoinverse of this Jacobian matrix provide the desired control commands for the joints. However, the blind application of this control strategy may drive the robot to unstable or unfeasible configurations for a standing pose. Therefore, we evaluate an stability condition of the robot configuration before applying the command. When stability is compromised we restrict the Visual Servoing to the head. Despite the linear nature of the approach we have not found convergence problems in practice. The approach overcomes the inaccuracies of the joint servo-motor controls and the image segmentation software. The actual implementation in the Aibo ERS-7 shows that the approach performs in real time when the seudoinverse is computed in the on-board processor of the robot. The real life experiments under controlled conditions have shown that the approach is highly robust to positioning of the ball in the field of view of the robot, it performs very fast and with very low final error, independently of the distance of the ball to the camera plane.

References

1. Altendorfer, R., Koditschek, D.E., Holmes, P.: Towards a factored analysis of legged locomotion models. In: IEEE International Conference on Robotics and Automation, 2003. Proceedings ICRA ’03, vol. 1, pp. 37–44 (2003)
2. Bretl, T., Lall, S.: Testing static equilibrium for legged robots. IEEE Trans. Robot. **24**(4), 794–807 (2008)
3. Chalup, S.K., Murch, C.L., Quinlan, M.J.: Machine learning with Aibo robots in the four-legged league of RoboCup. IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev. **37**(3), 297–310 (2007)
4. Cherubini, A., Giannone, F., Iocchi, L., Nardi, D., Palamara, P.F.: Policy gradient learning for quadruped soccer robots. Robot. Auton. Syst. **58**(7), 872–8787 (2010)
5. Colombo, C., Kruse, E., Sabatini, A.M., Dario, P.: Vision-based relative positioning through active fixation and contour tracking. In: Proceedings 2nd International Symp. on Intelligent Robotic Systems, SIRS’94, Grenoble, France, July, pp. 319–325 (1994)
6. Corke, P.: Visual Control of Robot Manipulators—A Review. Robotics and Automated Systems, vol. 7, pp. 1–31. World Scientific, Singapore (1993)
7. Sony Corporation: OPEN-R SDK model information for ERS-7 (2003)
8. Coste-Manière, E., Couvignou, P., Khosla, P.K.: Visual servoing in the task-function framework: a contour following task. J. Intell. Robot. Syst. **12**(1), 1–21 (1995)
9. Deakin, G.J.: Legged robots. Prod. Eng. **64**(9), 8 (1985)
10. Echegoyen, Z., d’Anjou, A., Graña, M.: Contribution to legged robot visual servoing. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) Knowledge-Based Intelligent Information and Engineering Systems, KES 2007. LNAI, vol. 4693, pp. 1179–1186. Springer, Berlin (2007)
11. Echegoyen, Z., d’Anjou, A., Graña, M.: Modeling a legged robot for visual servoing. In: Gervasi, O., Gavrilova, M.L. (eds.) Computational Science and Its Applications—ICCSA 2007. LNCS, vol. 4707, pp. 798–810. Springer, Berlin (2007)
12. Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Trans. Robot. Autom. **8**, 313–326 (1992)
13. Go, Y., Xiaolei, Y., Bowling, A.: Navigability of multi-legged robots. IEEE/ASME Trans. Mechatron. **11**(1), 1–8 (2006)
14. Graña, M., Torrealdea, F.J.: Hierarchically structured systems. Eur. J. Oper. Res. **25**, 20–26 (1986)
15. Hager, G.: The “xvision” system: a general purpose substrate for real-time vision-based robotics. In: Proceedings of the Workshop on Vision for Robotics, pp. 56–63 (1995)
16. Hill, J., Park, W.T.: Real-time control of a robot with a mobile camera. In: Proceedings of the 9th ISIR, Washington, DC, March, pp. 233–246 (1979)
17. Hoff, J., Bekey, G.A.: A cerebellar approach to adaptive locomotion for legged robots. In: 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA’97 Proceedings, July, pp. 94–100 (1997)
18. Hornby, G.S., Takamura, S., Yamamoto, T., Fujita, M.: Autonomous evolution of dynamic gaits with two quadruped robots. IEEE Trans. Robot. **21**(3), 402–410 (2005)
19. Hosoda, K., Kamado, M., Asada, M.: Vision-based servoing control for legged robots. In: IEEE International Conference on Robotics and Automation, vol. 4, pp. 3154–3159 (1997)
20. Hosoda, K., Miyashita, T., Takeuchi, S., Asada, M.: Adaptive visual servoing for legged robots-vision-cued swaying oflegged robots in unknown environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, pp. 778–784 (1997)
21. Hutchinson, S., Hager, G.D., Corke, P.I.: A tutorial on visual servo control. IEEE Trans. Robot. Autom. **12**(5), 651–670 (1996)
22. Kitano, H., Fujita, M., Zrehen, S., Kageyama, K.: Sony legged robot for RoboCup challenge. In: 1998 IEEE International Conference on Robotics and Automation, Proceedings, May, vol. 3, pp. 2605–2612 (1998)
23. Krasny, D.P., Orin, D.E.: Generating high-speed dynamic running gaits in a quadruped robot using an evolutionary search. IEEE Trans. Syst. Man Cybern., Part B, Cybern. **34**(4), 1685–1696 (2004)
24. Prajoux, R., de Martins, L.S.F.: A walk supervisor architecture for autonomous four-legged robots embedding real-time decision-making. In: Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems ’96, IROS 96, November, vol. 1, pp. 200–207 (1996)
25. Quinlan, M., Murch, C., Moore, T., Middleton, R., Li, L., King, R., Chalup, S.: The 2004 nubots team report. Technical report (2004)
26. Raibert, M.H., Tello, E.R.: Legged robots that balance. IEEE Expert **1**(4), 89–89 (1986)
27. Rives, P., Chaumette, F., Espiau, B.: Visual servoing based on a task function approach. In: Experimental Robotics I, Proceedings of the First International Symposium on Experimental Robotics, Montreal, Canada, June, pp. 412–428 (1990)
28. Rosen, C., Nitzan, D., Agin, G., Bavarsky, A., Gleason, G., Hill, J., McGhie, D., Park, W.: Machine intelligence research applied to industrial automation. Technical Report NSF Grant APR-75-13074, SRI Project 4391, 6th Report, SRI International, Menlo Park, CA, November 1976
29. Rosen, C., Nitzan, D., Agin, G., Bavarsky, A., Gleason, G., Hill, J., McGhie, D., Park, W.: Machine intelligence research applied to industrial automation. Technical report, 8th Report, SRI International, August 1978
30. Röfer, Th., Burkhard, H.-D., Düert, U., Homann, J., Göhring, D., Jüngel, M., Lötzsch, M., v. Stryk, O., Brunn, R., Kallnik, M., Kunz, M., Petters, S., Risler, M., Stelzer, M., Dahm, I., Wachter, M., Engel, K., Osterhues, A., Schumann, C., Ziegler, J.: Germanteam robocup 2003. Technical report, http://www.robocup.de/germanteam/GT2003.pdf, 2003
31. Samson, C., Le Borgne, M., Espiau, B.: Robot control: the task function approach. In: Oxford Engineering Science Series, vol. 22, 1st edn. Clarendon Press, Oxford University Press, Oxford (1991)
32. Sanderson, A.C., Weiss, L.E.: Image-based visual servo control using relational graph error signals. In: Proceedings of the IEEE International Conference on Cybernetics and Society, vol. 1, pp. 1074–1077 (1980)
33. Shirai, Y., Inoue, H.: Guiding a robot by visual feedback in assembling tasks. Pattern Recognit. **5**(2), 99–108 (1973)
34. Tira-Thompson, E.J.: Tekkotsu: A rapid development framework for robotics. Master’s thesis, Carnegie Mellon University, Pittsburgh, PA (2004)
35. Veloso, M., Uther, W., Fijita, M., Asada, M., Kitano, H.: Playing soccer with legged robots. In: 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, Proceedings, October, vol. 1, pp. 437–442 (1998)
36. Weiss, L.E.: Dynamic visual servo control of robots: an adaptive image-based approach. PhD thesis, Carnegie-Mellon University, April 1984
37. Wichman, W.M.: Use of optical feedback in the computer control of an arm. Technical report, Standford AI project, AI memo 55, August 1967
38. Yang, J.-M.: Fault-tolerant gaits of quadruped robots for locked joint failures. IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev. **32**(4), 507–516 (2002)

Z. Echegoyen has a degree in Computer Science from the University of the Republic of Uruguay and a Ph.D. from the Computer Science Department of the University of the Basque Country. He has working experience on factory automation and industrial robotics. He has been also working on Visual Servoing and Linked Multirobot systems modelling and control at the Computer Science Department of the University of the Basque Country. Nowadays he is working as a researcher at the Transport Unit of Tecnalia, since February 2010.

J.M. Lopez-Guede has a computer science degree from the Universidad del Pais Vasco (UPV/EHU) Assistant Professor at the Systems Engineering and Automatic Control Dept. and Ph.D. student at the Computer Science Dept., both of the Universidad del Pais Vasco. He is working on Linked Multirobot systems modeling, simulation and control using Computational Intelligence techniques.



B. Fernandez-Gauna has a computer science degree from the University of the Basque Country (UPV/EHU). He is currently a member of the Computational Intelligence Group, doing his PhD while performing as assistant professor. His current research interest are in robotics, specifically distributed control of multi-component robotic Systems.

M. Graña received the M.Sc. degree in Computer Science in 1982 and the Ph.D. degree in Computer Sciences in 1989, both from Universidad del Pais Vasco, Spain. His current position is Full Professor (Catedrático de Universidad) in the Computer Science and Artificial Intelligence Department of the Universidad del Pais Vasco, in San Sebastian. He is the head of the Computational Intelligence Group (Grupo de Inteligencia Computational) which has been awarded funding as a high performance university research group since the year 2001. Current research interests are in applications of Computational Intelligence to multicomponent robotic systems, specifically Linked Multicomponent Robotic Systems, medical image in the neurosciences, multimodal human computer interaction, remote sensing image processing, content based image retrieval, Lattice Computing, semantic modelling, data processing, classification and data mining.