# Modeling a legged robot for visual servoing

Zelmar Echegoyen, Alicia d'Anjou, and Manuel Graña *

Computational Intelligence Group, Dept. CCIA
Paseo Manuel de Lardiazábal, 1 20018 San Sebastian - Spain

**Abstract.** This article presents a contribution to the visual tracking of objects using all the degrees of freedom of an Aibo ERS-7 robot. We approach this issue in a principled way applying ideas of visual servoing. State of the art visual tracking solutions for this kind of robots inspired in the visual servoing approach either are restricted to the head effectors or they apply an inductive learning from experimental data approach to build up the kinematics matrix. In this work we take into account all the effectors which can affect the extrinsic parameters of the robot camera, and therefore in the captured image. We construct the robot kinematic matrix from its description. Visual servoing is performed computing the seudoinverse of this matrix.

## 1 Introduction

Visual servoing [1] is a technique for robot control which uses as a feedback signal the information extracted from the image sequences taken by one or several video cameras. In fact, it is defined as the control of the end-effector pose relative to a target object or set of features, for robotic manipulators, or the pose of the robot relative to some landmarks, in the case of mobile robotics. A major classification of visual servoing systems distinguishes position-based control from image-based control. In position- based control, the features extracted from the image are used to fit a geometric model of the target and the known camera model to estimate the pose of the robot relative to the target. Control feedback is computed trying to reduce errors in estimated pose space. In image-based servoing, control parameter values are computed on the basis of image features directly. We have chosen the image-based approach to reduce computational delay, avoiding the need for image interpretation and the errors due to sensor modeling and camera calibration. However the image-based approach imposes a linear approximation to obtain the control parameters. This is a significant simplification of the nonlinear and highly coupled robotic system whose effects must be evaluated by physical experimentation. Figure 1 illustrates the main feedback loop in image-based visual servoing with the Aibo.

In the RoboCup robot soccer matches some visual servoing approaches [2, 3] 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

* The MEC partially supports this work through grant DPI2006-15346-C03-03.

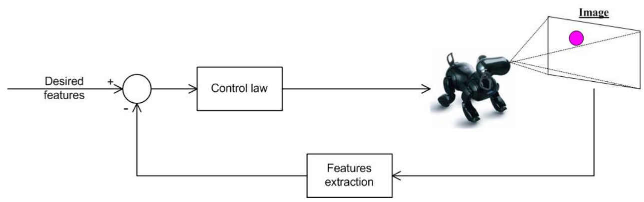



**Fig. 1.** Visual servoing feedbak loop

ball into the video image. The space in which the ball can be followed is greatly restricted by the robot body pose. Our approach is a systematic application of the visual servoing methodology, aimed at overcoming these limitations.

In this paper we address the problem of maintaining the playing ball in the center of the robot camera image. The only visual feature considered is the center of the ball region as identified in the image by the color detection routines implemented in the robot. We have profited from the CMU's SDK [4] and the SONY's SDK [5]. The image error is the distance in image space between the image center and the centroid of the blob corresponding to the ball. The image features considered are very naive when compared with recent works in other domains (i.e.: [6, 7]), however they are the current state of the art in the Aibo environment.

The control parameters are deduced applying the inverse of the linear approximation to the robot kinematic function given by the image Jacobian relative to the considered degrees of freedom. In this paper we detail the construction of the image Jacobian, starting from the geometric specifications of the Aibo robot. The blind application of the control parameter values given by the linear inverse kinematics may move the robot pose out of the useful configuration space, which we define as the set of standing stable positions. These positions are characterized by the relation between the support points and the robot's mass center. The support points are the points of contact of the robots limbs with the support surface. These points may correspond either to the leg ends or to the knees as illustrated in figure 2.

We introduce in section 2 the direct linear kinematics of the robot, then we develop the expression for the inverse kinematics in section 3 and we end up in 4 with some experimental results. The discussion of the physical implementation, the observed robot behavior and future work lines are given in 5

## 2 Direct Aibo Kinematics

We build the Aibo kinematics as a transformation from the ground supporting plane to the head coordinate system, composing the diverse transformations that correspond to the limbs and head degree of freedom. We start from the supporting points and go up to the head.

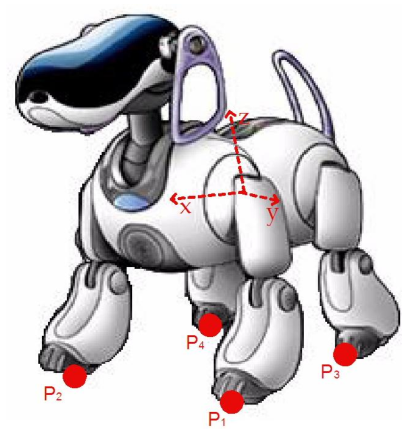



**Fig. 2.** Points of contact with the supporting surface

As illustrated in figure 2 the robot's feet and the knees are the possible robot support points therefore we need to be able to determine their 3D coordinates at any time.

### 2.1 Legs degrees of freedom

Each leg has three articulations, as shown in figure 4. The legs degrees of freedom are used indirectly towards the support points, so we introduce this concept.

**Support points** The support points are the points of the robot limbs that determine the plane where it is standing on. These points must be determined in the coordinate system of the dog. From the point of view of the center of the robot body the supporting plane apparently varies when the robot servos are affected when the physical reality is that the plane remains fixed and the robot changes its pose. We use the robot body center of mass because the Aibo possesses an inertial sensor than gives us feedback on the motion of this point.

Each leg has a unique support point that can be the foot as well as the knee, and, according to the restriction that the robot must be standing, at least three of the legs must have their supporting points in contact with the ground; therefore there are 32 possible support planes if we take into account all feasible combinations support points that may give us a standing configuration of the robot. In order to determine which combination of supporting points coincides with the physical supporting surface we obtain the plane equation for every possible combination.

For a given combination of support points we have the plane equation $\pi$ : $ax + by + cz + d = 0$, then we evaluate to which hemisphace belong the points that have not been taken into account to build the plane equation; if for any

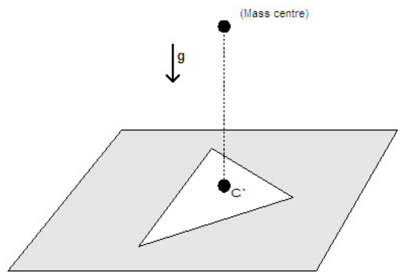



**Fig. 3.** Condition for supporting points on the ground plane

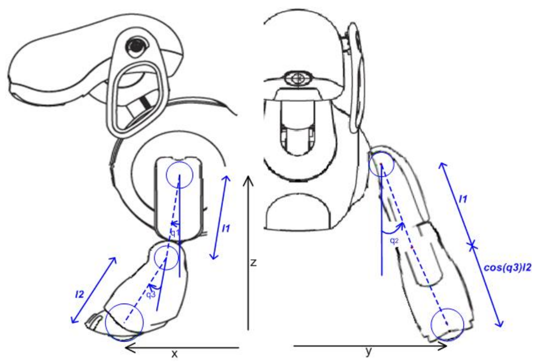



**Fig. 4.** Geometry of the leg articulations

one of them we find $(x_p, y_p, z_p)^T : ax_p + by_p + cz_p + d < 0$, it means that this support point is under the plane and therefore this plane is not the ground surface. Besides, in order for the robot to be standing in a stable pose, the projection of the body center of mass must lie inside of the triangle defined by the three supporting points in contact with the ground surface. This condition is illustrated in figure 3. Therefore, the search for the ground support points is guided by testing this condition on each triplet of leg supporting points. For those triplets that meet the condition, we fit the plane equation and test that the remaining supporting points remain above this plane.

**Feet and Knees positions** In order to obtain the ground supporting plane, is necessary to determine which are the supporting point coordinates for each leg in the reference space centered on the robot body center of mass. It is necessary to determine the positions of the feet and knees in function of the articulation states, given by their torsion angles.

We find the foot center position, for the front left leg, using the following coordinate system transformations.

$T_1$: Translation along de $z$-axis of length $-l_1$.

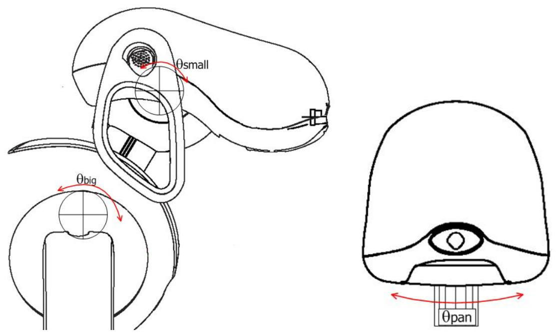



**Fig. 5.** Geometry of the head articulations

$R_1$: Clockwise rotation about $y$-axis by angle $q_1$.
$R_2$: Counterclockwise rotation about $x$-axis by angle $q_2$.
$R_1$: Clockwise rotation about $y$-axis by angle $q_3$.
$T_2$: Translation along de $z$-axis with length $-l_2$.
$T_l$: Translation along de $x$-axis with length $\frac{1}{2}l$, being $l$ the robot length.
$T_a$: Translation along de $y$-axis with length $\frac{1}{2}a$, being $a$ the robot width.

In homogeneous coordinates the transformation from the body center to the foot coordinate system can be described as the product of transformation matrices:

$$\begin{pmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{pmatrix} = (R_1.R_2.T_1.R_3.T_2).\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

This equation is valid for the robot front left leg; however, due to the symmetry of leg coordinate systems, only a few signs must be changed to get the positions of the other three leg's feet. In order to find the coordinates of each knee in the body reference system we only have to do the three first and the two last transformations used to determine the foot coordinates:

### 2.2 Head degrees of freedom

The Aibo ERS-7 has three degrees of freedom in the head. That introduces ambiguity in the control trajectories needed to track the ball trajectory. Figure 5 shows the two tilt degrees of freedom of the Aibo, denoted $\theta_{big}$ and $\theta_{small}$. The first head tilt degree of freedom corresponds to the neck base pivoting along part of the dog chest, while the second one allows the head to move vertically using as the rotation center the joint between the neck and the head. The third degree of freedom, called $\theta_{pan}$, allows a perpendicular rotation to the previous one, moving the head from side to side.

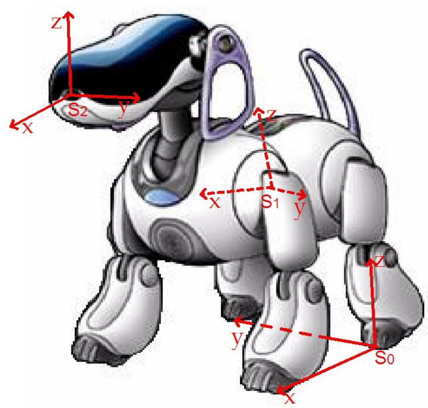



**Fig. 6.** Reference systems involved in the visual servoing

### 2.3 Image features

The stated goal is to bring the ball in the image center, so the target features are the image center coordinates and the observed features from the real world are the coordinates of the ball region center and its diameter. But these features must be expressed in terms of the robot degrees of freedom, in order to use the Jacobian to determine the feature sensitivity respect to each articulation positions changes.

**Coordinate reference systems** In order to obtain the ball position expressed in the $S_0$ system base it is necessary to obtain the transformation matrices between the different systems.

These reference systems are illustrated in figure 6.

*Transformation between $S_0$ and $S_1$* In order to define the coordinates changes between the base system, $S_0$, and the body system, $S_1$, we define the $S_0$ vectors in the system $S_1$, and then do the translation between them. So, we separate the transformation in rotation and translation, although it exists an scale component.

The entire transformation uses the supporting points positions: $r_i = (x_i, y_i, z_i, 1)^T$, $r_j = (x_j, y_j, z_j, 1)^T$, $r_k = (x_k, y_k, z_k, 1)^T$.We use the position point $r_i$ as the origin of $S_0$, and the vectors $\overrightarrow{r_i r_j}$ and $\overrightarrow{r_i r_k}$ as the two first vectors, and we built the third vector as the vectorial product of the two first.

So we built the rotational matrix, $R$, from the three vectors of $S_0$

$$R = \begin{pmatrix} \downarrow & \downarrow & \downarrow & 0 \\ r_j - r_i & r_k - r_i & \langle r_k - r_i, r_j - r_i \rangle & 0 \\ \downarrow & \downarrow & \downarrow & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

and we define the translational matrix from the origin of $S_1$ to the origin of $S_0$,

$$T = \begin{pmatrix} 1 & 0 & 0 & x_i \\ 0 & 1 & 0 & y_i \\ 0 & 0 & 1 & z_i \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

So, composing the two transformations we finally obtain the matrix change from $S_0$ to $S_1$,

$${}_{S_1}I_{S_0} = T \cdot R. \tag{3}$$

*Transformation between $S_1$ and $S_2$* The transformation between these systems can be done through the compositions of more elemental transformations. We will compose the transformations that go from the body system $S_1$, to the head system $S_2$.

The first transformation is a translation from the camera base to the top of the neck, $T_1$. Next, we have to rotate the head, taking into account the nod and pan articulations, we call this rotational matrix $R_1$. Then, we have to use the tilt articulation defining the rotational matrix $R_2$. Finally, the translation $T_2$, between the neck base and the body center, take the system to the $S_1$ origin.

The result of the matrix composition is the transformation between the systems $S_1$ and $S_2$

$${}_{S_2}I_{S_2} = T_2 \cdot R_2 \cdot R_1 \cdot T_1. \tag{4}$$

**Observed image features** The camera reference system is fixed to the robot head, and define the ball position according to the vision camera of the robot.

The observed image features, $c = (u, v)^T$, are determined by the ball position in the camera system, according to the following relation

$$\begin{pmatrix} u \\ v \end{pmatrix} = f(b_2) = \frac{\lambda}{x_{b_2}} \begin{pmatrix} y_{b_2} \\ z_{b_2} \end{pmatrix}. \tag{5}$$

The features are expressed in terms of the ball position in the system $S_2$, but as we had supposed the ball was fixed respect to $S_0$, we could obtain the features expressed in function of the head robot articulations and the support points positions, using the ball position in $S_0$ and the transformation between $S_0$ and $S_2$.

$$\begin{pmatrix} u \\ v \end{pmatrix} = f({}_{S_2}I_{S_0} . b_0) \tag{6}$$

## 2.4 Feature Jacobian matrix

Now we will construct the Jacobian matrix that relates the variations of the diverse degrees of freedom of the robot with the variations in the image plane.

**Dependence on the features** Deriving the equation 5 we get the following relation:

$$\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \begin{pmatrix} \frac{-\lambda . y_p}{x_p^2} & \frac{\lambda}{x_p} & 0 & 0 \\ \frac{-\lambda . z_p}{x_p^2} & 0 & \frac{\lambda}{x_p} & 0 \end{pmatrix} \begin{pmatrix} \delta x_{b_2} \\ \delta y_{b_2} \\ \delta z_{b_2} \\ 0 \end{pmatrix} \tag{7}$$

We call $J_{cb}$ the Jacobian matrix of the equation 7. Then $J_{cb}$ defines a linear transformation from variations of the positions of the ball in $S_2$ into variations of the image features.

$$\Delta c \simeq J_{cb} \cdot \Delta b \tag{8}$$

**Dependence on the target object** We saw that the ball position in the camera system could be expressed as a function of the support points and the head robot articulations, equation 6. By deriving this equation we get the Jacobian matrix that relates the variations in the image ball position with the variations in the support points positions and in the head articulations

$$J_{br} = \frac{\delta({}_{S_2}I_{S_1} \circ_{S_1} I_{S_0})}{\delta r} b_0 \tag{9}$$

Using the chain rule, we rewrite the equation 9:

$$J_{br} = [\frac{\delta({}_{S_2}I_{S_1})}{\delta r} \circ ({}_{S_1}I_{S_0}) + ({}_{S_2}I_{S_1}) \circ \frac{\delta({}_{S_1}I_{S_0})}{\delta r}] b_0 \tag{10}$$

As ${}_{S_2}I_{S_1}$ is a function of $r_{head}$ (head articulations) and ${}_{S_1}I_{S_0}$ is a function with parameter $r_{legs}$ (support points positions), we rewrite 10:

$$J_{br} = [\frac{\delta({}_{S_2}I_{S_1})}{\delta r_{head}} \circ ({}_{S_1}I_{S_0}) + ({}_{S_2}I_{S_1}) \circ \frac{\delta({}_{S_1}I_{S_0})}{\delta r_{legs}}] b_0 \tag{11}$$

The dependence between the variations in the ball position and the variations in the head degree of freedoms and in the legs positions can be summarized by:

$$\Delta b \ \simeq \ J_{br} \cdot \Delta r \tag{12}$$

**Dependence on the support points** The next step is obtaining a linear transformation between the variations of the legs degrees of freedom and the ground support points coordinates in the body reference system.

First we observe that according to which part of the leg is in contact with the ground there are two possible jacobean matrices, one for the foot ($J_{p_i}$ ) an another for the knee ($J_{r_i}$). We model the changes in the foot and the knees coordinates according to the degrees of freedom variations, using the Jacobians as follows:

$$\Delta p_i \simeq Jp_i \cdot \Delta J_i \tag{13}$$
$$\Delta r_i \simeq Jr_i \cdot \Delta J_i \tag{14}$$

Being $\partial J_{3i}$, $\partial J_{1i}$ and $\partial J_{2i}$ the value of the variations of the degrees of freedom in leg $i$.

Composing with the support points Jacobian at every moment, we obtain the following jacobian matrix:

$$\begin{pmatrix} \partial P_1 \\ \partial P_2 \\ \partial P_3 \\ \partial P_4 \end{pmatrix} = \begin{pmatrix} M_1 & 0 & 0 & 0 \\ 0 & M_2 & 0 & 0 \\ 0 & 0 & M_3 & 0 \\ 0 & 0 & 0 & M_4 \end{pmatrix} \begin{pmatrix} \partial J_1 \\ \partial J_2 \\ \partial J_3 \\ \partial J_4 \end{pmatrix} \tag{15}$$

The jacobian matrix receives the name $J_{p\theta}$,where $M_i$ is:

• $Jp_i$, if the support point for the leg $i$ is the foot.

• $Jr_i$, if the support point for the leg $i$ is the knee.

• $Zero$ (the matrix with all the elements equal 0) if this leg has not a lean point on the plane.

The dependence of the support ground points on the limb's degrees of freedom is summarized as follows:

$$\Delta P \ \simeq \ J_{p\theta}.\Delta J \tag{16}$$

**Dependence on the robot articulations** The following matrix relates the variations in the generic support points, $\delta r$, with the variations in the true legs support points and the free leg support point, $\delta p$. We call $\delta r_h$ and $\delta p_h$ the head articulation variations in order to define $r$ and $p$.

$$\begin{pmatrix} \delta p_h \\ \delta p_1 \\ \delta p_2 \\ \delta p_3 \\ \delta p_4 \end{pmatrix} = \begin{pmatrix} Id & 0 & 0 & 0 \\ 0 & M_{1i} & M_{1j} & M_{1k} \\ 0 & M_{2i} & M_{2j} & M_{2k} \\ 0 & M_{3i} & M_{3j} & M_{3k} \\ 0 & M_{4i} & M_{4j} & M_{4k} \end{pmatrix} \begin{pmatrix} \delta r_h \\ \delta r_i \\ \delta r_j \\ \delta r_k \end{pmatrix} \tag{17}$$

We call $J_{pr} \in \mathbb{R}^{15\times 12}$ the Jacobian matrix of equation 17, and we define this matrix as a composition of matrixes of size $3 \times 3$. The first row starts with the identity and the rest of matrixes are null. So, $\delta r_h$ and $\delta p_h$ are equals.

To the rest of matrixes:

$M_{xy} = Id$, if the leg $x$ has the support point $y$,
$M_{xy} = 0$, if the leg $x$ has a support point different from $y$,
$M_{xy} = \alpha_{xy}$, if the leg $x$ has not a generic support point.

So, if the leg $x$ has a generic support point, then the row $x$ has a identity matrix and three null matrix; while for the free leg we get the following row

$$\begin{pmatrix} 0 \ \alpha_i \ \alpha_j \ \alpha_k \end{pmatrix} \tag{18}$$

This coefficient matrixes define the relation between the free leg variations and the variations in the rest of legs. This coefficients can be used to generate stability and advance behaviors. If the first element of the row is not 0, then we can make the free leg position depend on the head articulations variations.

The dependence between the variations in the legs articulations with the supporting points positions and the head articulations variations can be summarized in the following equation

$$\Delta p \simeq J_{pr} \cdot \Delta r \tag{19}$$

Defining the Jacobian matrix $J_{r\theta}$, as the product $J^{+}_{pr} \circ J_{p\theta}$, we get the following dependence relation between the variations of support points positions and robot articulations,

$$\Delta r \simeq J_{r\theta} \cdot \Delta\theta \tag{20}$$

**Full Jacobian matrix** Finally, to obtain the full Jacobian matrix that models the dependence of the image features on the diverse degrees of freedom of the robot we must compose the previous transformations given by equations (8), (12), (19) and (16).

$$\Delta c = [(J_{cb} \circ J_{br}) \circ J_{r\theta}].\Delta\theta \tag{21}$$

We denote the composite matrix $J_{c\theta} = (J_{cb} \circ J_{br}) \circ J_{r\theta}$.

## 3 Inverse Kinematics

The goal of the stated visual servoing problem is to determine the instantaneous variation at each robot degree of freedom that will be needed to bring the ball center to the image plane center.

In order to determine the velocity at each robot degree of freedom we should obtain the inverse of the $J_{c\theta}$ matrix in equation 21. However, this is not possible because the matrix is not invertible. As we have more degrees of freedom than image features, the problem is overconstrained, because there are not sufficient features to determine the movements in an unique way.

The general solution is to use the pseudoinverse of $J^{+}_{c\theta}$, thus obtaining the following approximation to the instantaneous robot degree of freedom velocity vector:

$$\dot{\theta} = J^{+}_{c\theta}\dot{c} + (I - J^{+}_{c\theta}J_{c\theta})n \tag{22}$$

Being $n$ an arbitrary vector of $R^{15}$.

In general, $(I - J^{+}_{xj}J_{xj})n \neq 0$, and all the vectors of the form $(I - J^{+}_{xj}J_{xj})n$ belong to the kernel of the transformation associated to $J_{c\theta}$. This solution minimizes the norm

$$\left\| \dot{x} - (J_{c\theta})\dot{\theta} \right\| \tag{23}$$

As our objective is to bring the ball to the image center, we can neglect the second order term, so our estimation of the velocity vector will be

$$\dot{\theta} = J^+_{c\theta}\dot{c} \tag{24}$$

This solution does not take into account the restriction of keeping the distances between supporting points constant. So, we need to determine how the variations in the supporting points positions, as a consequence of the robot joint movement, affect the distances between them. Let us consider the vector of distances:

$$l = \begin{pmatrix} \|l_1\| \\ \|l_2\| \\ \|l_3\| \end{pmatrix} = \begin{pmatrix} \|r_i - r_j\| \\ \|r_j - r_k\| \\ \|r_k - r_i\| \end{pmatrix} \tag{25}$$

Differencing $l$ we get the Jacobian matrix $J_{lr}$ that relates these changes in position with the change in relative distances between supporting points:

$$J_{lr} = \begin{pmatrix} 0 & \frac{\delta l_1}{\delta r_i} & \frac{\delta l_1}{\delta r_j} & \frac{\delta l_1}{\delta r_k} \\ 0 & \frac{\delta l_2}{\delta r_i} & \frac{\delta l_2}{\delta r_j} & \frac{\delta l_2}{\delta r_k} \\ 0 & \frac{\delta l_3}{\delta r_i} & \frac{\delta l_3}{\delta r_j} & \frac{\delta l_3}{\delta r_k} \end{pmatrix} \tag{26}$$

Finally this dependence is summarized in the following equation:

$$\Delta l \simeq J_{lr} \cdot \Delta r \tag{27}$$

Now we may combine the following Jacobian matrices: $J_{rc}$, $J_{lr}$ and $J_{r\theta}$ to compute the variations on the robot articulations that make the image features converge to the desired features, keeping constant the distances between the supporting points. To fulfill this restriction, the $\Delta r$ vector must belong to the kernel of the transformation associated to $J_{lr}$. The following equation ensures that $\Delta r$ belongs to the kernel of $J_{lr}$.

$$\Delta r = [(I - J^+_{lr}J_{lr})\{(I - J^+_{lr}J_{lr})J^+_{rc}\}^+]\Delta c \tag{28}$$

As our final objective is to compute the movements at the robot joints, we multiply by the seudoinverse of $J_{r\theta}$. The control of the robot joints bringing the ball center to the image center is given by the following iterative rule:

$$\Delta\theta = \alpha J^+_{r\theta}(I - J^+_{lr}J_{lr})\{(I - J^+_{lr}J_{lr})J^+_{rc}\}^+\Delta c, \tag{29}$$

where $\alpha$ is the control gain that modulates the application of the rule, the experimental results show the sensitivity of the approach to the values of this gain.

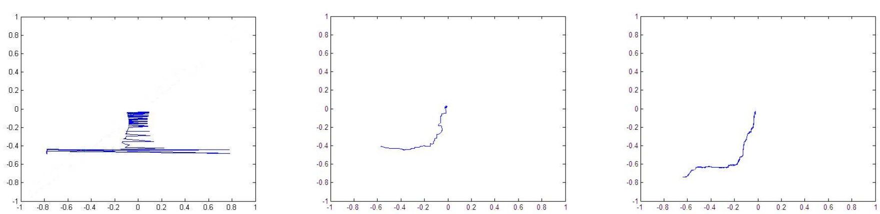
This equation is unrestricted and may drive the robot into unstable configurations, that is, to articulation configurations that fall out of the region of stable poses in configuration space. Stable poses are characterized by the existence of a triplet of ground support points which fulfill the condition illustrated in figure 3. When this does not happen, or the projection point is too close to the triangle boundary, we restrict the visual servoing to the head degrees of freedom, using the transformation $S_0(I)S_2$ to construct a reduced Jacobian $M_h$ that relates the image features to the head degrees of freedom. Its seudoinverse gives the control for the head degrees of freedom. This reduced approach has already been applied in [2, 3] :

$$d\theta = M_h^+ .dr \tag{30}$$

## 4 Experimental results

We tested our approach on the Aibo robot. In the real life experiments the starting position of the ball is in the lower left corner of the image. The experimental results show the sensitivity of our control rule convergence to the gain parameter. Figure 7 show some instances of trajectories of the ball center in the image plane obtained for different settings of the gain parameter. It can be observed that $\alpha = 1$ produces a fast convergence followed by strong oscillations. Lower vales equivalent to damped control, show slower but oscillation free convergence. Very small values result in control commands below the precision of the robot joints.

The experiments have also show that there is some sensitivity to the initial robot pose. In order to prevent the robot to escape its standing configuration region, if we detect that the robot tries to perform a joint movement that risks escaping the standing configuration region, we block all motion of the body joints and restrict the calculation of the motion of the head joints, as described above.



(a) (b) (c)

**Fig. 7.** Trajectories in the image plane of the ball center for different values of the gain parameter (a) $\alpha = 1$, (b) $\alpha = 0.5$, (c) $\alpha = 0.3$

## 5 Conclusion

We have developed the visual servoing for the whole set of degrees of freedom of the Aibo 7 following a principled approach. From the geometrical description of the robot we have constructed the full Jacobian matrix that linearizes the functional dependence of the image plane viewed by the robot camera on the robot degrees of freedom. The seudoinverse of this Jacobian matrix provide the desired controls. The blind application of this control strategy may lead the robot to unstable or unfeasible configurations for a standing pose. Therefore, we test the stability of the robot configuration. When it is compromised we restrict the visual servoing to the head. The implementation shows that the approach gives real time response when the seudoinverse is computed in the onboard processor of the robot. We are actually performing the real time experiments and collecting performance information.

## References


1. Hutchinson, S. Hager, G. D. Corke, P. I. “A tutorial on visual servo control” IEEE Transactions on Robotics and Automation, 12 (5):651-670
2. M. Quinlan, C. Murch, T. Moore, R. Middleton, L. Li, R King, and S. Chalup, ”The 2004 NUbots Team Report”, 2004, http://robots.newcastle.edu.au/publications/NUbotFinalReport2004.pdf.
3. Th. Röfer, H.-D. Burkhard, U. Düffert, J. Hoffmann, D. Göhring, M. Jüngel, M. Lötzsch, O. v. Stryk, R. Brunn, M. Kallnik, M. Kunz, S. Petters, M. Risler, M. Stelzer, I. Dahm, M. Wachter, K. Engel, A. Osterhues, C. Schumann, and J. Ziegler. GermanTeam RoboCup 2003. Technical report, 2003, http://www.robocup.de/germanteam/GT2003.pdf.
4. www.tekkotsu.org.
5. Sony Corporation, ”OPEN-R SDK Model Information for ERS-7”, 2003.
6. Pomares, J. Torres, F. Movement-flow-based visual servoing and force control fusion for Manipulation Tasks in unstructured environments IEEE trans. Systems, Man and Cybernetics, Part C 35(1):4-15
7. Garcia-Aracil, N. Malis, E. Aracil-Santonja, R. Perez-Vidal, C. Continuous visual servoing despite the changes of visibility in image features IEEE Trans. Robotics 21(6):1214-1220