Vestibular Calibration of a Humanoid Robot's Head

José Santos and Alexandre Bernardino

***Abstract*— Many modern robotic devices are characterized by complex mechanics and a large number of sensors and degrees of freedom. Moreover due to miniaturization, some joints may lack absolute sensors which leads to unknown initial configurations at startup. To control such systems a reliable initialization and calibration of sensors and actuators is required. Classical calibration methods rely on precise mechanical adjustments and are often arduous and prone to errors. In this paper we propose a sensor based methodology to calibrate and initialize a humanoid robot head roll-pitch-yaw angles using an inertial (vestibular) sensor fixed at its top. The calibration is done during the initialization procedure exploiting the characteristics of the head kinematics. Results are shown both in simulation and in the real system comparing batch and incremental approaches.**

I. INTRODUCTION

The calibration of robotic devices with many sensors and actuators is often an arduous task. Classical calibration methods require precise mechanical adjustments and parameter tuning done by experts. With the growing complexity of robotic devices, the development of automated and self-calibration methods will significantly impact on the usability and ease of maintenance of such systems.

The availability of multiple sensors in modern robots opens opportunities for the development of sensor based calibration methods. Such methods exploit the information provided by the sensors together with some prospective movements executed by the robot to self-calibrate the sensorimotor relationships between the sensors and the actuators. As an outcome, the estimation of the unknown configuration at startup is often possible to obtain. In a previous work [6] we have developed visual based methods for calibrating the eyes pan-tilt degrees of freedom of a humanoid stereo head. In this paper we use an inertial sensor emulating the vestibular sense, to calibrate and initialize the neck pan-tilt-swing angles (see Fig. II). Using both methodologies it is possible to fully calibrate the humanoid head.

Despite only showing an application to a humanoid stereo head, the method is more general. In particular it is applicable to the class of kinematics chains with three serial rotation joints orthogonally arranged having an inertial sensor at the end of the chain. We propose two different approaches, one based on the Newton's Method for non-linear equations and the other based on the Broyden's method [2]. Their relative efficiency is discussed and the one which presented the best experimental results was implemented on a real humanoid robot: the iCub.

II. PROBLEM FORMULATION

The humanoid head system considered in this work has six degrees of freedom: neck pitch (tilt), neck roll (swing), neck yaw (pan), eyes tilt, eyes version and eyes vergence, as shown in figure II. In this work, we are solely interested in the head movements: tilt, swing and pan.

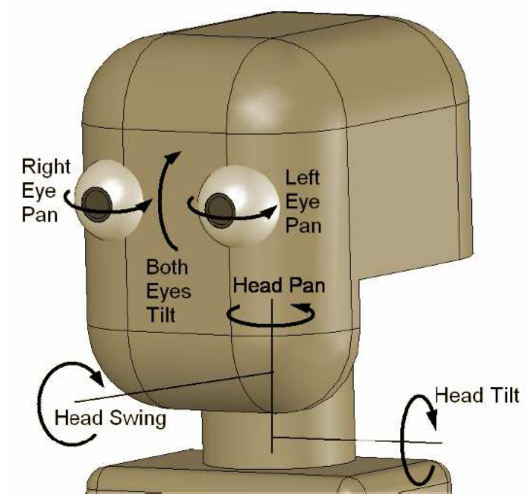


Fig. 1. Illustration of the iCub's head degrees of freedom.

It is important to establish a reference coordinate frame, which will be denoted by $\{0\}$. This reference coordinate frame corresponds to the body coordinate frame and is defined by the local vertical and by the rotation of the body along this axis. Considering identical reference coordinate frames for all joints in the canonical state, the rotation matrix representing the head's orientation with respect to the body reference frame depends on the tilt, swing and pan angular displacements, and is given by [1]:

$$
\begin{aligned}
{}^0\mathrm{R}_3 &= {}^0\mathrm{R}_1 \cdot {}^1\mathrm{R}_2 \cdot {}^2\mathrm{R}_3 && (1)\\
&= \mathrm{ROT}_y(\theta_t) \cdot \mathrm{ROT}_x(\theta_s) \cdot \mathrm{ROT}_z(\theta_p) && (2)\\
&= \begin{bmatrix} c_t & 0 & s_t \\ 0 & 1 & 0 \\ -s_t & 0 & c_t \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_s & -s_s \\ 0 & s_s & c_s \end{bmatrix} \cdot \begin{bmatrix} c_p & -s_p & 0 \\ s_p & c_p & 0 \\ 0 & 0 & 1 \end{bmatrix} && (3)
\end{aligned}
$$

with $c_t = cos(\theta_t)$, $s_t = sin(\theta_t)$, $c_s = cos(\theta_s)$, $s_s = sin(\theta_s)$, $c_p = cos(\theta_p)$ and $s_p = sin(\theta_p)$.

The robot head has an inertial sensor at the end of the pan joint. The inertial sensor unit is composed of accelerometers and rate-gyros for the three axis and a magnetometer to measure the azimuth with respect to a earth fixed co-ordinate system. The inertial unit is thus able to calculate the orientation between the sensor-fixed coordinate system, denoted by $\{S\}$, and the earth fixed co-ordinate system, denoted by $\{G\}$, which is defined as a right handed Cartesian coordinate system with:

- $\hat{X}$ positive when pointing to the local magnetic North.
- $\hat{Y}$ according to the right handed coordinates (West).
- $\hat{Z}$ positive when pointing up.

Note that this co-ordinate system may not correspond to the body coordinate system (that only happens if the inertial sensor is placed facing North). Moreover, the sensor fixed co-ordinate system does not correspond to the head fixed co-ordinate system (above denoted by $\{3\}$), because of the way the sensor is installed in iCub's head. This two coordinate systems are related acording to the following rotation matrix:

$$^{3}\mathrm{R}_S = \mathrm{ROT}_z(\pi) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore, one can say that the inertial sensor outputs the following rotation matrix:

$$^{G}\mathrm{R}_S = {}^{G}\mathrm{R}_0 \cdot {}^{0}\mathrm{R}_3 \cdot {}^{3}\mathrm{R}_S \tag{4}$$

As already mentioned, the angular displacements given by the motor encoders are measured with respect to the position in which the robot is turned on. Thus, using the information provided by the motor differential encoders (the joint angle displacements) and the information provided by the inertial sensor($^{G}\mathrm{R}_S$), our goal is to align iCub's head with its body. That is, iCub's head should be put in a position such that $^{3}\mathrm{R}_0 = I_3$.

III. The Proposed Methodology

Considering equation 4 and having performed the required calculations, one can state that:

$$\left({}^{G}\mathrm{R}_S\right)_{31} = \mathrm{r}_{31}^{GS} = s_t \cdot c_p - c_t \cdot s_s \cdot s_p \tag{5}$$

$$\left({}^{G}\mathrm{R}_S\right)_{32} = \mathrm{r}_{32}^{GS} = -s_t \cdot s_p - c_t \cdot s_s \cdot c_p \tag{6}$$

The goal is to place the iCub's head in a position such that $^{0}\mathrm{R}_3 = I_3$. In such a position the following equalities must be verified:

$$\mathrm{r}_{31}^{GS} = 0 \tag{7}$$

$$\mathrm{r}_{32}^{GS} = 0 \tag{8}$$

It is important to note that, since $^{G}\mathrm{R}_S$ is a rotation matrix, if equations 7 and 8 are verified then the following equalities must also hold:

$$\mathrm{r}_{13}^{GS} = 0 \tag{9}$$

$$\mathrm{r}_{23}^{GS} = 0 \tag{10}$$

$$\mathrm{r}_{33}^{GS} = 1 \tag{11}$$

In order to put the iCub's head in a position in which equations 7 and 8 hold, we only need to change the tilt and swing angular displacements. This problem is one of solving a system of nonlinear equations (the number of equality conditions is equal to the number of variables):

$$r(\theta_t, \theta_s) = \begin{bmatrix} r_{31}^{GS}(\theta_t, \theta_s) \\ r_{32}^{GS}(\theta_t, \theta_s) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{12}$$

Note that these constraints only align the head Z axis with gravity. The azimuth (pan angle) is still undetermined and will be computed with additional constraints.

A popular way to solve this kind of problems is to use Newton's method for nonlinear equations [2]. Newton's method defines a linear model $M_k(\Delta\theta)$ of $r(\theta + \Delta\theta)$ in the following way:

$$r(\theta + \Delta\theta) = r(\theta) + J(\theta) \cdot \Delta\theta \tag{13}$$

where $J(\theta)$ denotes the jacobian of r evaluated in θ.

Newton's method in its pure form chooses the step $\Delta\theta$ to be the vector for which $M_k(\Delta\theta) = 0$, that is:

$$\Delta\theta = -J(\theta)^{-1} \cdot r(\theta) \tag{14}$$

However, in this case, since r is not known, the information required for computing the jacobian is not given. As such, we propose two different strategies to compute the initial tilt and swing angular displacements, respectively: θ_t^0 and θ_s^0.

A. A Systematic Approach

One way of solving this problem is to express r_{31}^{GS} and r_{32}^{GS} in the following way:

$$\begin{aligned} \mathrm{r}_{31}^{GS}(\alpha,\beta) =& \sin(\theta_t^0 + \alpha)\cos(\theta_p^0) \\ & - \cos(\theta_t^0 + \alpha)\sin(\theta_s^0 + \beta)\sin(\theta_p^0) \end{aligned} \tag{15}$$

$$\begin{aligned} \mathrm{r}_{32}^{GS}(\alpha,\beta) =& - \sin(\theta_t^0 + \alpha)\sin(\theta_p^0) \\ & - \cos(\theta_t^0 + \alpha)\sin(\theta_s^0 + \beta)\cos(\theta_p^0) \end{aligned} \tag{16}$$

where α and β are suitably chosen angle displacements of the tilt and swing joints. The above equations can be rewritten as expressed below:

$$\begin{aligned} \mathrm{r}_{31}^{GS}(\alpha,\beta) =& a_1 \sin(\alpha) + a_2\cos(\alpha) + a_3 \sin(\alpha)\sin(\beta) \\ & + a_4 \sin(\alpha)\cos(\beta) + a_5\cos(\alpha)\sin(\beta) \\ & + a_6 \cos(\alpha)\cos(\beta) \end{aligned} \tag{17}$$

$$\begin{aligned} \mathrm{r}_{32}^{GS}(\alpha,\beta) =& b_1 \sin(\alpha) + b_2\cos(\alpha) + b_3 \sin(\alpha)\sin(\beta) \\ & + b_4 \sin(\alpha)\cos(\beta) + b_5\cos(\alpha)\sin(\beta) \\ & + b_6 \cos(\alpha)\cos(\beta) \end{aligned} \tag{18}$$

Since the values of a_1,...,a_6 and b_1,...,b_6 depend on the values of θ_t^0, θ_s^0 and θ_p^0, they are unknowns. However, in order to determine their values, one has simply to collect the values of r_{31}^{GS} and r_{32}^{GS} in six different points (α_1, β_1),..., (α_6, β_6) and then solve the linear system of equations obtained by writing equations 17 and 18 for each one of these points. In practice, since the information provided by the inertial sensor is affected by noise, one should collect more than six points and then use the Pseudo-Inverse matrix method.

After rewriting equation 12 in terms of α and β, one obtains:

$$r(\alpha, \beta) = \begin{bmatrix} r_{31}^{GS}(\alpha,\beta) \\ r_{32}^{GS}(\alpha,\beta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{19}$$

which is known (as explained: this amounts to solve a system of linear equations). So Newton's method for nonlinear equations can now be applied to determine the angular displacements α^* and β^* which solve equation 19. Clearly, the initial tilt and swing angular displacements measured with respect to the body reference frame are given by:

$$\theta_t^0 = -\alpha^* \tag{20}$$

$$\theta_s^0 = -\beta^* \tag{21}$$

B. An Incremental Approach

Another way of addressing this problem consists in using Broyden's method [2]. Broyden's method is a secant method: it constructs its own approximation of the Jacobian, updating it at each iteration so that it mimics the behavior of the true Jacobian J over the step just taken.

The requirement that the approximate Jacobian should mimic the behavior of the true Jacobian can be specified as follows. Let s_k denote the step from θ_k to θ_{k+1} and let y_k denote the corresponding change in r, that is:

$$s_k = \theta_{k+1} - \theta_k \tag{22}$$

$$y_k = r(\theta_{k+1}) - r(\theta_k) \tag{23}$$

Broyden's method requires that the updated Jacobian approximation B_{k+1} to satisfy the following equation, which is known as the secant equation:

$$y_k = B_{k+1} \cdot s_k \tag{24}$$

The secant equation ensures that B_{k+1} and $J(x_{k+1})$ have similar behavior along direction s_k. Broyden's method corresponds to the following update:

$$B_{k+1} = B_k + \frac{(y_k - B_k \cdot s_k) \cdot s_k^T}{s_k^T \cdot s_k} \tag{25}$$

The Broyden update makes the smallest possible change to the Jacobian (as measured by the Euclidean norm: $\|B_k - B_{k+1}\|_2$) that is consistent with the secant equation, which can be formally stated as:

$$B_{k+1} \in \underset{B \,:\, y_k = B \cdot s_k}{\arg\min} \|B - B_k\| \tag{26}$$

The specification of the algorithm is presented below.

Algorithm 1 Broyden's Method

Choose θ_0 and a nonsingular initial
    Jacobian approximation B_0;
for $k = 0, 1, 2, \cdots$ **do**
  Calculate a solution $\Delta\theta_k$ to the linear equations:
    $B_k \cdot \Delta\theta_k = -r(\theta_k)$
  $\theta_{k+1} \Leftarrow \theta_k + \Delta\theta_k$
  $s_k \Leftarrow \theta_{k+1} - \theta_k$
  $y_k \Leftarrow r(\theta_{k+1}) - r(\theta_k)$
  Obtain B_{k+1} from formula 25
end for

After applying Broyden's algorithm it is reasonable to expect that the tilt and swing displacements are almost zero. So:

$$\theta_t^0 = -\sum_{k=1}^{n} \Delta\theta_t^k \tag{27}$$

$$\theta_s^0 = -\sum_{k=1}^{n} \Delta\theta_s^k \tag{28}$$

C. Computing the Initial Pan

Having applied one of these two approaches to compute θ_t^0 and θ_s^0, one can easily compute the initial pan displacement θ_p^0 by means of equations 5 and 6:

$$s_p^0 = \frac{b \cdot r_{31}^0 - a \cdot r_{32}^0}{a^2 + b^2} \tag{29}$$

$$c_p^0 = \frac{r_{31}^0 - b \cdot s_p^0}{a} \tag{30}$$

with $a = s_t$ and $b = -c_t \cdot s_s$. Therefore:

$$\theta_p^0 = atan2(s_p^0, c_p^0) \tag{31}$$

Here, only the information corresponding to the initial position is being used in order to compute θ_p^0. However, both the approaches presented in this work collect information corresponding to several positions while computing θ_t^0 and θ_s^0. These data could be used to determine θ_p^0 more precisely, using, for instance, a Weighted Least Squares estimator [4]. Observe that positions closer to the zero present a greater signal-to-noise ratio and, thus, should be assigned smaller weights.

IV. Experimental Results

Both Newton's Method and Broyden's Method (as described in algorithm 1) were implemented in Matlab in order to assess the way each one converges when applied to the problem of aligning the iCub's head with its body. We assume that, initially, each joint angle (tilt, swing and pan) is reasonably close to 0 with respect to the body reference frame (every joint angle is assumed to be lower than 30°).

A. Inertial Sensor Noise Characterization

As was stated in section II, the inertial sensor outputs a rotation matrix, ${}^G\mathrm{R}_S$, which describes the orientation of the sensor fixed co-ordinate system, $\{S\}$, with respect to the earth fixed co-ordinate system, $\{G\}$. Naturally, the information provided by this sensor includes noise. Therefore, in order to simulate it properly one needs to characterize the noise variance.

We have evaluated the sample variance of the rotation matrix provided by the inertial sensor in a wide range of positions. In each position we recorded 100 samples and then computed the sample variance. The maximum variance registered was 0.00334233. Hence, in each of the simulations presented in this section we shall assume a zero-mean gaussian white noise with variance 0.0034.

B. Matlab Simulations

For both methods we performed several simulations with different initial conditions and measured the error between the ground truth position (zero angles) and the output of the algorithms. For each initial condition, ten experiments were performed. In each experiment, only ten iterations of the algorithms are executed, since we experimentally verified that when the algorithms converge, they typically converge quickly. Nevertheless, in some experiments the algorithms did not converge; therefore, only the successful trials are considered when computing the average error between the ground truth position and the ouput of the algorithm (we consider a trial to be successful if the error corresponding to each one of the joint angles is less than 0.5rad). Tables I and II show the average errors expressed in radians as well as the number of non-convergent trials out of ten.

TABLE I
APPLICATION OF NEWTON'S METHOD TO FOUR DISTINCT INITIAL CONFIGURATIONS

Initial Configuration	θ_{tilt}	θ_{swing}	θ_{pan}	#Failures
$\frac{\pi}{6}, \frac{\pi}{6}$	0.0509	0.0629	0.1067	1
$\frac{\pi}{6}, \frac{\pi}{12}$	0.0493	0.0431	0.0706	0
$\frac{\pi}{6}, \frac{\pi}{12}$	0.0436	0.0902	0.1460	1
$\frac{\pi}{12}, \frac{\pi}{12}$	0.0593	0.0298	0.1233	1

TABLE II
APPLICATION OF NEWTON'S METHOD TO FOUR DISTINCT INITIAL CONFIGURATIONS

Initial Configuration	θ_{tilt}	θ_{swing}	θ_{pan}	#Failures
$\frac{\pi}{6}, \frac{\pi}{6}$	0.0812	0.0309	0.1334	1
$\frac{\pi}{6}, \frac{\pi}{12}$	0.0778	0.0709	0.1404	1
$\frac{\pi}{6}, \frac{\pi}{12}$	0.0177	0.0228	0.0807	0
$\frac{\pi}{12}, \frac{\pi}{12}$	0.0222	0.0179	0.0692	2

The simulations presented in figures 2 to 5 illustrate the way θ_{tilt} and θ_{swing} are changed in each iteration of Broyden's Method.

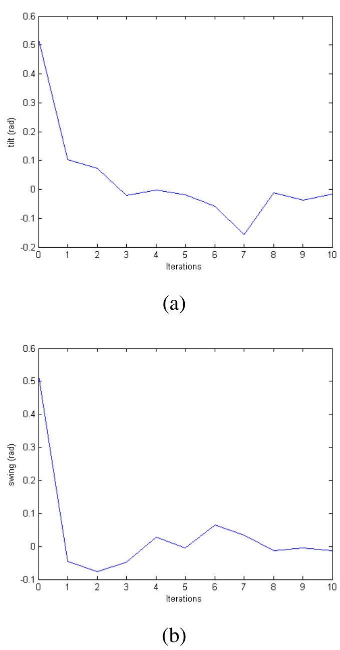


Fig. 2. Evolution of the tilt (top) and swing (bottom) angles along the iterations of the Broyden's method. Initial configuration: $\theta_{tilt} = \frac{\pi}{6}$ $\theta_{swing} = \frac{\pi}{6}$

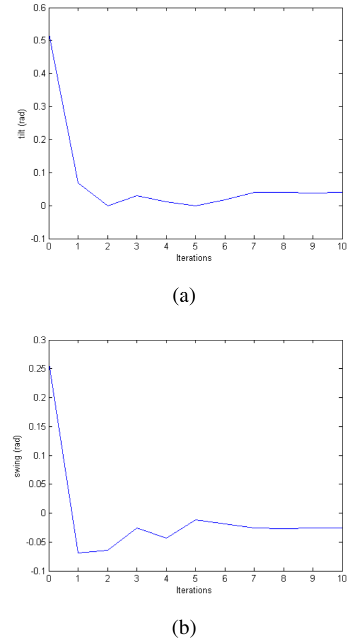


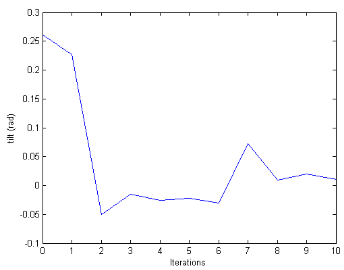
Fig. 3. Evolution of the tilt (top) and swing (bottom) angles along the iterations of the Broyden's method. Initial configuration: $\theta_{tilt} = \frac{\pi}{6}$ $\theta_{swing} = \frac{\pi}{12}$

TABLE III
EVOLUTION OF THE INERTIAL SENSOR ORIENTATION MATRIX ALONG THE ITERATIONS OF THE METHOD. INITIAL CONFIGURATION:
$\theta^0_{tilt} = 0.7561rad \quad \theta^0_{swing} = 0.3047rad \quad \theta^0_{pan} = 0.5927rad$

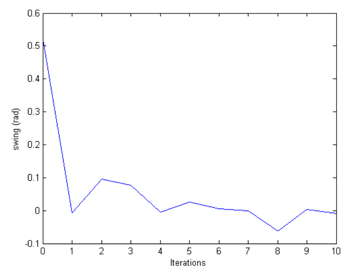
Iterations	r_{31}	r_{32}	r_{13}	r_{23}	r_{33}
0	0.448124	-0.565588	-0.365259	-0.622327	0.692311
1	0.404702	0.03108	0.220109	-0.34103	0.91392
2	0.030612	0.262968	0.250975	0.084264	0.964319
3	-0.138035	0.059402	-0.008302	0.150045	0.988644
4	-0.064814	-0.097917	-0.116638	0.013573	0.993082
5	0.054952	-0.052626	-0.019847	-0.073453	0.997101
6	0.042871	0.023935	0.041518	-0.02621	0.998794
7	0.003065	0.041944	0.038273	0.017432	0.999115
8	-0.029368	-0.002641	-0.016464	0.024462	0.999565
9	0.003777	-0.005103	-0.002637	-0.005775	0.99998

C. Implementation on iCub

Broyden's algorithm was successfully implemented on the iCub. Considering the nature of the problem at hand, it is quite difficult to evaluate its results in practice, since the real zero is not known. Nevertheless, we can illustrate the application of the algorithm by showing how the entries of the rotation matrix provided by inertial sensor change during the corresponding application. When the head of the robot is aligned with its body, the orientation matrix provided by the inertial sensor must be a rotation about the Z axis. So, the entries r_{31}, r_{32}, r_{13} and r_{23} must be zero and entry r_{33} must be 1. Table III presents the evolution of these entries when Broyden's algorithm is applied to the real robot. Naturally, it is only possible to estimate the initial joint angles after applying the algorithm and assuming that the head is then aligned with the body.

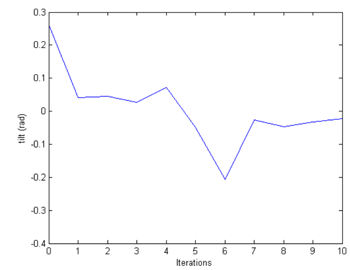


(a)

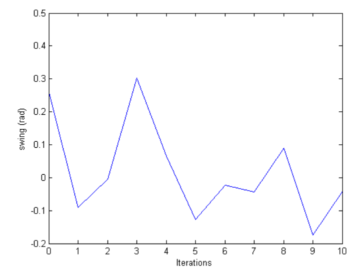


(b)

Fig. 4. Evolution of the tilt (top) and swing (bottom) angles along the iterations of the Broyden's method. Initial configuration: $\theta_{tilt} = \frac{\pi}{12}$ $\theta_{swing} = \frac{\pi}{6}$



(a)



(b)

Fig. 5. Evolution of the tilt (top) and swing (bottom) angles along the iterations of the Broyden's method. Initial configuration: $\theta_{tilt} = \frac{\pi}{12}$ $\theta_{swing} = \frac{\pi}{12}$

V. Conclusions and Future Work

In this paper we have presented a method to automatically calibrate the neck joints of a humanoid robot to its default configuration, using measurements from the inertial sensor placed in the top of the head. The method is suitable to be used in systems lacking absolute sensing in their actuators' joints.

Two alternative methods were presented: (i) a batch method requiring the acquisition of inertial measurements on several different head configurations before computing the solution; and (ii) an incremental method computing the solution while performing the prospective movements. Results show that both methods provide good results in most cases. However, in some circumstances, mainly when the head configuration is taken to the limits of the workspace, the methods may not converge to the solution. Whereas these situations can be diagnosed online using the incremental method, the batch method is not able to deal with these cases.

The Broyden's method, due to its ability to diagnose online algorithm non-convergence and to intrinsically avoid the joint limits is the method of our choice. In terms of speed, we have empirically shown that it converges to values close to zero in less than ten iterations. Errors measured in the real robot and in simulation trials with realistic noise conditions are very low and confirm the practical utility of the proposed method.

In future work we will aim at integrating visual and vestibular information to perform the full calibration of the head eyes and neck sub-systems. Additionally we aim at performing a more extensive analysis of the convergence basin of the algorithms.

References

[1] R. Beira, M. Lopes, M. Praça, J. Santos-Victor, A. Bernardino, G. Metta, F. Becchi, and R. Saltarn. Design of the robot-cub (icub) head. *Proc. IEEE International Conference on Robotics and Automation, ICRA 2006*, May 2006.

[2] G. Broyden. A class of methods for solving non-linear simultaneous equations. *Mathematics of Computation*, 19:577–593, October 1965.

[3] J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, first edition edition, 1955.

[4] L. Ljung. *System Identification - Theory for the User*. Prentice Hall, second edition edition, 1999.

[5] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, second edition edition, 1999.

[6] B. Tworek, A. Bernardino, and J. Santos-Victor. Visual self-calibration of pan-tilt kinematic structures. *Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, ROBOTICA 2008*, April 2008.