Autonomous learning of 3D reaching in a humanoid robot

Francesco Nori, Lorenzo Natale, Giulio Sandini
Italian Institute of Technology
Via Morego 30, Genova, ITALY
{*francesco.nori, lorenzo.natale, giulio.sandini*}*@iit.it*

Giorgio Metta
University of Genoa and Italian Institute of Technology
Via Morego 30, Genova, ITALY
giorgio.metta@iit.it

***Abstract*— In this paper, we describe the implementation of a precise reaching controller on an upper-torso humanoid robot. The solution we propose does not rely on prior models of the kinematic structure of either the arm or the head. A learning strategy enables the robot to acquire the required sensory-motor transformations. After learning the robot is able to precisely reach for a visually identified object in the 3-dimensional space. In this technique we use the fixation point (represented in the head joints motor space) as a reference frame to code the position of the object and to represent the eye-to-hand Jacobian matrix. This strategy successfully deals with the kinematic redundancy of the structure and constraints the dimensionality of the problem.**

***Index Terms*— Visual servoing, reaching, learning, Jacobian, humanoid robotics.**

I. Introduction

To reach for a visually identified object the robot has to solve an inverse kinematic problem. This problem has been studied extensively in the robotic literature and many solutions exist. None of these solutions is easy to implement in practice. The main limitations are that they usually require some knowledge of the robot kinematics and that the camera parameters are either available "a priori" or estimated by means of a specific calibration procedure. The problem becomes even more complicated for humanoid robots which are often characterized by a large number of degrees of freedom (DOF) and redundancies in the arm and head. The kinematic of the robot in these systems becomes a complex function whose estimation is not, in practice, a trivial task.

In general solving the reaching task requires a sensory-motor map transforming the position of the target in the visual domain into the arm joint vector which solves the task. Extensive literature describe calibration procedures for retrieving at least part of the parameters which characterize this mapping [1], [2]. These techniques can be quite accurate but require the robot to operate in somewhat structured environment, typically involving a known object (or a calibration grid), and a calibrated hand pose sensor. Some procedures have been proposed to relax part of these assumptions. In particular [3] avoids the need for a calibrated object, whereas [4] exploits specific kinematic constraints to avoid an explicit measure of the hand pose.

Depending on how we decide to encode the target position (and the relative sensory-motor transformation) these calibration procedures might not be necessary. Assuming that the robot is fixating the target, the joints of the head implicitly encode the position of the target in the world. In this case a simpler approach could be to directly map the joint vector defining the current head posture to the arm joint vector corresponding to the configuration of the arm that brings the hand to the fixation point. We call this map a *forward "motor-motor" map* because it is a direct link between the motor angle of the head and those of the arm [5], [6].

Reaching can be performed by inverting this motor-motor map. An extension of this approach to redundant manipulators has been proposed in [7].

The procedure to learn the motor-motor map is straightforward and can be easily acquired in a "motor babbling" phase in which the robot moves the arm randomly while maintaining fixation on the hand. The limitation of these approaches is that reaching is intrinsically ballistic and does not use visual feedback. The precision of forward map is learned poses intrinsic bounds on the accuracy of the reaching task.

This problem can be solved by using a visual servoing control schema (for a review: [8]). These techniques use the Jacobian matrix to control the arm so that the visual distance between the hand and the target is progressively reduced to zero. [9]–[11] describe procedures to estimate the Jacobian matrix of a robot. Unfortunately these techniques do not suggest how to deal with highly redundant systems. The Jacobian matrix, in fact, is a function of both the arm and the head joints, and its estimation becomes quickly hard as the number of joints increases.

A somewhat intermediate approach is described in [12]. In this case the robot uses stereo vision to compute the position of the hand in the 3D space; reaching is then accomplished by inverting the forward mapping between the arm joints and the 3D position of the hand. This solution employs a visuomotor map, which naturally maps vision to motor space. Indeed from the forward map the authors derives the *eye-to-hand visual Jacobian* of the arm which allows the robot to drive the hand to the target with arbitrary precision. To deal with the degrees of freedom problem the authors

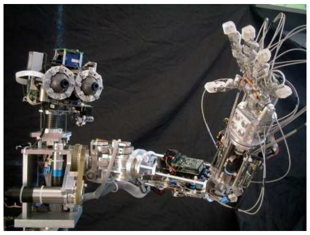

Fig. 1. The humanoid robot James.

Fig. 2. Two typical images taken from the two cameras mounted on the eyes of the robot (resolution 320×240). The coordinates of the target (the blue ball) on the image planes will be denoted u_r, v_r, and u_l, v_l (respectively right and left cameras).

propose an approximate solution where the head joint vector is substituted for by the output of the robot gyroscope. The drawback of this approach is that it requires prior calibration of the visual system and no relative movement between the cameras. In practice this solution forces the robot to maintain the eyes stationary in a fixed position with respect of the head.

The technique we propose in this paper integrates the open loop and the closed loop approaches, as it was previously proposed in [13]. The main idea consists in learning a suitable (motor-motor) map to direct the hand close to the fixated object. The closed loop controller is then activated as soon as visual feedback of the hand becomes available. Together these strategies enable the robot to initiate a reaching action whether or not sight of the hand is available (e.g. when the hand is out of view or in the dark). The closed loop controller allows taking advantage of the visual feedback to arbitrarily reduce the positioning error. Differently from [13], we propose a methodology to autonomously learn both the open-loop motor-motor map and the eye-to-hand Jacobian matrix. The method we propose does not rely on any prior information about the robot kinematic structure nor it requires camera calibration or solving the explicit 3D problem.

II. The robotic setup

The experiments described in this paper were carried out on the robot James (see figure 1). The head is equipped with two eyes, which can pan and tilt independently (4 DOFs), and is mounted on a 3 DOFs neck. The arm has 7 DOFs: three of them are located in the shoulder, one in the elbow and three in the wrist. The hand has five fingers and is under-actuated with a total of 20 joints controlled by 8 motors thus resulting in 8 DOFs.

The head structure has a total of 7 degrees of freedom, actuated by 8 motors. Four of these motors are used to independently actuate the pan and tilt movements of the left and right eyes. Even though the eyes can be moved independently, our strategy was to couple their movements so to achieve a more human-like motion. We used common tilt α_t^c, vergence α_v^d and version α_v^c.

The neck has three degrees of freedom, denoted θ_y, θ_p and θ_r for yaw, pitch and roll respectively.

To summarize, the variables relevant to understand the remainder of the paper are: the head joints $\mathbf{q}_{head}$ = $\begin{bmatrix}\alpha_t^c & \alpha_v^d & \alpha_v^c & \theta_y & \theta_p & \theta_r\end{bmatrix}^\top \in \mathbb{R}^6$ and the first four arm joints (3 d.o.f. shoulder and elbow) denoted $\mathbf{q}_{arm} \in \mathbb{R}^4$.

III. Gaze control

In this section we describe how we control the gaze of the robot to fixate a visual target. Let (u_r, v_r) and (u_l, v_l) be the coordinates of the target on the right and left image plane respectively (see Figure 2). Directing gaze toward the target consists in moving the neck and the eyes so as to obtain $u_r = 0$, $v_r = 0$, $u_l = 0$, $v_l = 0$. Let us define the vector $\mathbf{u}_{target} = \begin{bmatrix}u_r & u_l & v_l\end{bmatrix}^\top \in \mathbb{R}^3$ corresponding to the target location in the image planes[1]. Our control task can be redefined as the problem of controlling $\mathbf{u}_{target}$ to zero.

A. Head redundancy w.r.t. the task

Clearly, the above task specification ($\mathbf{u}_{target} \to 0$) does not constrain all the head degrees of freedom (we are imposing $m = 3$ constraints but we have $n = 6$ free variables available: we remain with $n - m = 3$ additional degrees of freedom). To solve this "redundancy problem" we imposed three additional constraints which are required to be satisfied when fixating the target: $\alpha_v^c = 0$, $\alpha_t^c = 0$ and $\theta_r = 0$. Mathematically the above constraints are implemented as follows:

$$\begin{cases}\dot{\alpha}_v^c &= K_p(u_l + u_r) \\ \dot{\theta}_y &= K_y \alpha_v^c\end{cases}, \quad \begin{cases}\dot{\alpha}_t^c &= K_t(v_l + v_r) \\ \dot{\theta}_p &= K_r \alpha_t^c\end{cases}, \tag{1}$$

Vergence is instead controlled separately:

$$\dot{\alpha}_v^d = K_p(u_l - u_r). \tag{2}$$

Finally, the neck roll degree of freedom θ_r is maintained fixed, i.e. $\theta_r^d = 0$.

The proposed control strategy allows us to asymptotically fixate the target ($u_l \to 0$, $v_l \to 0$, $u_r \to 0$ which implies $v_r \to 0$) while also guaranteeing an asymptotically straight gaze ($\alpha_v^c \to 0$, $\alpha_t^c \to 0$). Moreover, by choosing a suitable value for the gains K_p, K_y, K_t and K_r it is possible to achieve an asymptotic behavior with the eyes moving rapidly on the target and the neck following the eye movement with a slower movement.

[1] We here assume that the cameras are horizontally aligned, i.e. $u_r = v_r$

IV. REACHING

In our formulation the problem of reaching consists in moving the robot arm so as to reach with the hand an observed target. Classically, reaching passes trough the estimation of the cartesian position of a target $\tilde{\mathbf{x}}_{target} \in \mathbb{R}^3$ (with respect to a robot centered reference frame) using the stereo camera system. Given that the head is moving, we have $\tilde{\mathbf{x}}_{target} = F(\mathbf{q}_{head}, \mathbf{u}_{target})$ for a suitably defined function F. In practice, the estimation of this function is a delicate calibration procedure consisting in two steps: a camera calibration and a kinematic calibration [1], [12].

Given the target position, reaching can be performed by means of the forward kinematic map G, which gives the hand cartesian position $\tilde{\mathbf{x}}_{hand}$ as a function of the robot arm configuration: $\tilde{\mathbf{x}}_{hand} = G(\mathbf{q}_{arm})$. The combination of the maps F and G can be used to perform open and closed loop target reaching[2].

In this paper we tackle the reaching problem with a different approach, originally proposed in [5] and here extended to the case of a redundant head and arm. Instead of representing the target position in the cartesian space $\tilde{\mathbf{x}}_{target}$, we use a representation in a space $\mathbf{x}_{target}$ homogeneous to the previous one but strictly related to the robot motor space. With this representation no camera calibration is needed and a fully autonomous learning can be performed.

A. Open Loop Reaching

Suppose that the robot is tracking a target as described in Section III. Ideally, after fixation is achieved, we have $\mathbf{q}_{head} = \begin{bmatrix}\theta_y & \theta_p & 0 & \alpha_v^d & 0 & 0\end{bmatrix}^\top \in \mathbb{R}^6$. Since there exists a one to one mapping between the three dimensional position of the target $\tilde{\mathbf{x}}_{target}$ and the three non-zero variables θ_y, θ_p and α_v^d, we can define $\mathbf{x}_{target} = \begin{bmatrix}\theta_y & \theta_p & \alpha_v^d\end{bmatrix}^\top \in \mathbb{R}^3$. This new variable $\mathbf{x}_{target} \in \mathbb{R}^3$ uniquely codes the spatial position of the target in a way that resembles a three dimensional polar representation. In particular θ_y and θ_p code respectively *azimuth* and *elevation*, while *distance* is substituted with α_v (the *vergence* angle).

If the robot tracks the hand, the same subset of the head joint space can be used to code the spatial location of the hand: $\mathbf{x}_{hand} = \begin{bmatrix}\theta_y & \theta_p & \alpha_v^d\end{bmatrix}^\top \in \mathbb{R}^3$. Under these assumptions, the *forward mapping* $f_{arm} : \mathbb{R}^4 \longrightarrow \mathbb{R}^3$ relates the arm configuration $\mathbf{q}_{arm}$ with the position of the hand $\mathbf{x}_{hand}$:

$$\mathbf{x}_{hand} = f_{arm}(\mathbf{q}_{arm}), \qquad f_{arm} : \mathbb{R}^4 \longrightarrow \mathbb{R}^3. \tag{3}$$

In the next section we show how a neural network could be trained to approximate the arm forward mapping (Eq. 3).

Suppose now that the map has already been learned and that the robot is fixating a target to be reached. Formally the problem can be formulated as determining the value of $\mathbf{q}_{arm}$ which solves the following optimization problem:

$$\min_{\mathbf{q}_{arm}} (J) = \min_{\mathbf{q}_{arm}} \|\mathbf{x}_{hand} - \mathbf{x}_{target}\|^2, \tag{4}$$

where $\mathbf{x}_{target}$ is measured from the encoders of the head, while $\mathbf{x}_{hand}$ is computed from $\mathbf{q}_{arm}$ through Eq. (3). Given the redundancy of the arm kinematics the minimization (4) has infinite solutions. We constrained the problem by forcing one of the joints, for example joint number 2 (one of the shoulder joints), to remain as close as possible to a predefined value q_{20}:

$$\min_{\mathbf{q}_{arm}} (J_c) = \min_{\mathbf{q}_{arm}} \left[\|\mathbf{x}_{hand} - \mathbf{x}_{target}\|^2 + (q_{arm,2} - q_{20})^2\right]. \tag{5}$$

The optimization of (5) can be performed numerically using various algorithms. In our implementation, we used the downhill simplex method as implemented in [14].

1) Learning the open loop reaching: To learn the forward motor-motor map of Eq. (3) we programmed the robot to perform random movements with the arm. During this "exploratory" phase the robot tracked the hand, and collected samples of the form: $\left(\ \mathbf{q}_{arm}^i, \mathbf{x}_{hand}^i\ \right)_{i=0,1...}$. A neural network was then trained to learn the relation $\mathbf{x}_{hand} = \hat{f}_{arm}(\mathbf{q}_{arm})$, which approximates Eq. (3).

In the experiment reported in this paper we collected a data set of about 2890 samples that we divided in a training set ($N_{train} = 2168$ samples) and a test set ($N_{test} = 725$ samples). The neural network we employed was the Receptive Field Weighted Regression model proposed by [15]. This network implements an online learning method, meaning that a learning step is performed every time a new sample is presented to the network. All samples in the training set were shown to the network in a random order. After each training step the performance of the network was validated on the whole test set, by computing the Mean Squared Error (*MSE*) between each sample in the test set, $\mathbf{x}_{hand}^i$, and the corresponding network output, $\hat{\mathbf{x}}_{hand}^i$. The plot in figure 3 shows the trend of the error on the test set during learning. At the end of the training the network reached the performance of $MSE = 5.7$ (with $STD = 10.4$).

B. Closed Loop Reaching

By measuring the visual distance between the hand and the target, we can improve the reaching accuracy by implementing a closed control loop. The underlying idea consists in

[2]Given an observed target $\mathbf{u}_{target}$ with the head in the position $\mathbf{q}_{head}$, the open loop reaching is the result of:

$$\mathbf{q}_{arm} = \arg\min_{q} \|G(\mathbf{q}) - F(\mathbf{q}_{head}, \mathbf{u}_{target})\|.$$

When the hand is visible on the image plane,residual (image plane) errors can be reduced implementing a visual servoing controller:

$$\dot{\mathbf{q}}_{arm} = -\frac{\partial G}{\partial \mathbf{q}}^{\#} \frac{\partial F}{\partial \mathbf{u}} (\mathbf{u}_{hand} - \mathbf{u}_{target}),$$

where $A^{\#}$ is the Moore-Penrose pseudo-inverse of the matrix A.

performing a preliminary (open loop) reaching movement and then refining the action by visually correcting any residual error. A classical closed loop implementation based on the forward model (3) would be the following:

$$\dot{\mathbf{q}}_{arm} = -\left[\frac{\partial f_{arm}}{\partial \mathbf{q}_{arm}}\right]^{\#}(\mathbf{x}_{hand} - \mathbf{x}_{target}),$$

where $A^{\#}$ denotes the Moore-Penrose pseudo-inverse of a matrix A. However, if we are fixating the target, we do not have direct access to $\mathbf{x}_{hand}$ which, instead, requires to fixate the hand. A possible solution consists in directly measuring $\mathbf{x}_{hand}$ by temporarily shifting the robot attention to the hand. Alternatively, our idea is to estimate the eye-to-hand Jacobian matrix which relates arm velocities $\dot{\mathbf{q}}_{arm}$ with hand velocities in the image plane $\dot{\mathbf{u}}_{hand} = \left[\begin{array}{ccc} \dot{u}_r & \dot{u}_l & \dot{v}_l \end{array}\right]^{\top}$:

$$\dot{\mathbf{u}}_{hand} = \tilde{\mathbf{J}}\left(\mathbf{q}_{arm}, \mathbf{q}_{head}\right)\dot{\mathbf{q}}_{arm}, \tag{6}$$

where $\tilde{\mathbf{J}} \in \mathbb{R}^{3\times 4}$ depends on both the configuration of the arm and the head. In practice, assuming sufficiently small arm movements $\Delta\mathbf{q}_{arm}$, we can use the following approximation:

$$\Delta\mathbf{u}_{hand} = \tilde{\mathbf{J}}\left(\mathbf{q}_{arm}, \mathbf{q}_{head}\right)\Delta\mathbf{q}_{arm}, \tag{7}$$

where $\Delta\mathbf{u}_{hand} = [\Delta u_r,\ \Delta u_l,\ \Delta v_l]^{\top}$ is the image plane displacement resulting from the arm movement $\Delta\mathbf{q}_{arm}$. Due to the additional constraints posed by the head tracker, we showed that only a subset of $\mathbf{q}_{head}$, $\mathbf{x}_{target}$, is sufficient to uniquely identify the position of the head, so we can rewrite equation (7) as:

$$\Delta\mathbf{u}_{hand} = \tilde{\mathbf{J}}\left(\mathbf{q}_{arm}, \mathbf{x}_{target}\right)\Delta\mathbf{q}_{arm}, \qquad \tilde{\mathbf{J}} \in \mathbb{R}^{3\times 4}. \tag{8}$$

Moreover, after the preliminary open loop reaching movement, we know that $\mathbf{x}_{target} = \hat{f}(\mathbf{q}_{arm})$ so that Eq. (8) can be further simplified to:

$$\Delta\mathbf{u}_{hand} = \mathbf{J}\left(\mathbf{q}_{arm}\right)\Delta\mathbf{q}_{arm}, \qquad \mathbf{J} \in \mathbb{R}^{3\times 4} \tag{9}$$

where $\mathbf{J}$ depends only on the arm joint configuration $\mathbf{q}_{arm}$.

Using (9) as an approximation for (6), we can implement the following closed loop controller:

$$\dot{\mathbf{q}}_{arm} = -k \cdot \mathbf{J}^{\#}(\mathbf{q}_{arm})\mathbf{u}_{hand}, \qquad \mathbf{J}^{\#} \in \mathbb{R}^{4\times 3}, \tag{10}$$

where k is an arbitrary positive scalar. This control law minimizes the error $e = \|\mathbf{u}_{hand} - \mathbf{u}_{target}\|^2$ and asymptotically drives the visual position of the hand $\mathbf{u}_{hand}$ to the visual position of the target $\mathbf{u}_{target} = 0$.

1) Learning the Closed Loop Reaching: To implement (10) we need to evaluate $\mathbf{J}$ (actually its pseudo-inverse $\mathbf{J}^{\#}$).

As described in Section IV-A.1, the robot moves the arm randomly, while maintaining gaze on the hand. At the end of each movement j the arm is in a configuration $\mathbf{q}^j_{arm}$, while the eyes are fixating the hand ($\mathbf{u}_{hand} \approx 0$) with a straight gaze (the head tracker has reached convergence). Each arm configuration corresponds to a different value of $\mathbf{J}_j = \mathbf{J}\left(\mathbf{q}^j_{arm}\right)$. Now the robot inhibits the head tracker and performs a sequence m of small arm movements $\Delta\mathbf{q}^k_{arm}$ which perturb $\mathbf{u}_{hand}$ of small amounts $\Delta\mathbf{u}^k_{hand}$:$\left(\ \Delta\mathbf{u}^k_{hand}, \Delta\mathbf{q}^k_{arm}\ \right)_{k=0,1\ldots,m}$. All m perturbations $\Delta\mathbf{u}^k_{hand}$ and $\Delta\mathbf{q}^k_{arm}$ are linearly related through $\mathbf{J}_i$ as described in Eq. (8). From these m observations we can derive a least squares estimation of $\mathbf{J}_j$ from which, in turn, we can compute the pseudo-inverse $\mathbf{J}^{\#}_j$.

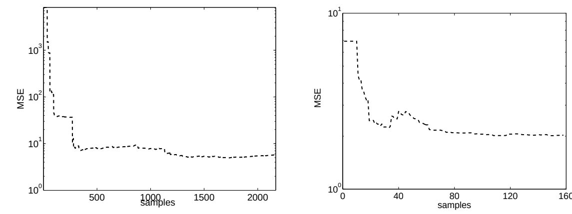

Fig. 3. Left: learning of the arm forward function. Right: learning the arm jacobian. The plots represent the *MSE* on the test set during learning. See text for more details

Re-iterating this procedure leads to the collection of a series of examples: $\left(\ \mathbf{q}^j_{arm}, \mathbf{J}^{\#}_j\ \right)_{j=0,1\ldots}$. An approximation $\hat{\mathbf{J}}^{\#}$ of $\mathbf{J}^{\#}$ is finally obtained by training a neural network:

$$\mathbf{g}\left(\mathbf{q}_{arm}\right), \qquad g : \mathbb{R}^4 \longrightarrow \mathbb{R}^{12}, \tag{11}$$

whose output components are the coefficients of $\hat{\mathbf{J}}^{\#} \in \mathbb{R}^{4\times 3}$.

We report here the result of a learning session. The robot explored 210 different arm positions $\mathbf{q}^j_{arm}$ randomly distributed within a region of the workspace. In each of these positions the robot executed $m = 10$ perturbations $\Delta\mathbf{q}^k_{arm}$ and estimated an example $\mathbf{J}^{\#}_j$ for the neural network. We trained the neural network on a subset of $N_{train} = 158$ elements (training set); each sample was shown to the network only once and then discarded. Following each training step, we evaluated the performance of the network by computing *MSE* on the remaining $N_{test} = 52$ elements (test set). At the end of the training the error on the test set was *MSE* $= 2$ (*STD* $= 7.1$). Figure 3 reports the plot of the error during learning.

V. Results

In this section we report the results of the experiments we carried out to quantify the performance of the reaching movements. Following the proposed strategy, in order to reach for the target we first need to fixate it, i.e. $\mathbf{u}_{target} = 0$. Clearly, the image plane distance $\|\mathbf{u}_{hand} - \mathbf{u}_{target}\| = \|\mathbf{u}_{hand}\|$ can be used as a rough estimate of the reaching precision, i.e. of the Cartesian distance between the target to be reached and the position of the hand.

A. Open Loop

The first attempt to reach the target consists in using the learned motor-motor map (3) and the strategy (5) Clearly, if the forward kinematic function (3) were perfectly represented

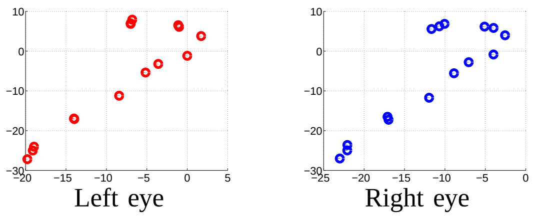

Fig. 4. Open loop image plane errors $\mathbf{u}_{hand}$ for different choices of the redundant variable q_{20}. On the horizontal axis u_r and u_l; vertical axis v_r and v_l (always in pixels).

and if the target were reachable, then, after (5), we would have $\mathbf{x}_{hand} = \mathbf{x}_{target}$, which implies $\|\mathbf{u}_{hand} - \mathbf{u}_{target}\| = 0$. In practice, the model (3) cannot exactly represent the system kinematic[3]. Therefore, even tough we can find $\mathbf{q}_{arm}$ such that $\mathbf{x}_{hand} = \hat{f}_{arm}(\mathbf{q}_{arm})$ it is not guaranteed that after the movement execution $\|\mathbf{u}_{hand} - \mathbf{u}_{target}\| = 0$. Figure 4 shows the image plane errors after the execution of the open loop movement. The plot has been obtained by fixating a target and performing a series of open loop movements. Each open loop movement was different because (5) was solved by choosing a different value q_{20}.

B. Closed Loop

Residual visual errors can be reduced by a visual closed loop control strategy (10), started immediately after the open loop phase. Relatively weak conditions on the learned Jacobian [16] guarantee the convergence of the image plane errors $\mathbf{u}_{hand}$ to zero, and therefore the convergence of the hand $\mathbf{x}_{hand}$ to the target $\mathbf{x}_{target}$. Figures 5 shows how the hand is actually driven to the exact image center in both the image planes. Accuracy is improved but at the cost of a slower execution speed (see Figure 5 right); faster executions couldn't be obtained by increasing the control loop gains, due to the frame rate (thirty milliseconds) and the delays in the visual processing (hand localization and tracking). Finally, it is important to notice the quasi-linearity of the path followed by the hand during the closed loop phase. This linearity denotes a good accuracy of the learned Jacobian.

C. Superimposed Open and Closed Loop

Finally, we tested an alternative control strategy based on activating the closed loop phase immediately after the hand becomes visible on both image planes. Practically, a feedforward control performs the open loop part of the reaching movement and it does not require the hand to be visible in the image plane. The feedback control instead corresponds to the closed loop part of the movement and can be activated when the hand has been localized in both the image planes (see Figure 6).

[3]Part of the representational errors are related to the representation of the kinematic function, in this case the so called Receptive Field Weighted Regression model. Part are due to the mechanical plays and backlash of the mechanical structure.

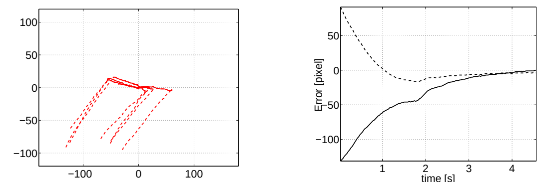

Fig. 5. Left: movement of the hand on the left image plane (320×240) during the execution of different reaching actions. Solid line: closed loop. Dashed trace: open loop. Right: time response of the closed loop and open loop strategy on the left image plane. Solid line: u_l. Dashed lines: v_l.

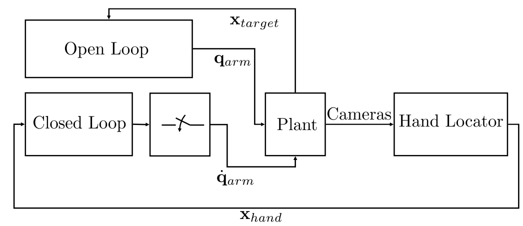

Fig. 6. Scheme of the simultaneous open and closed loop control.

With this control strategy when the open and closed loop controllers are active, the system receives position and velocity control simultaneously[4]. A comparison between this control strategy and the one proposed in Section IV-B is given in Figure 7. The image plane movement (Figure 7) is much more regular resulting and the execution time is reduced (right part of Figure 7).

D. Null space movement

In order to validate the quality of the Jacobian estimation, we tested the effects of a null space movement on the primary

[4]In our system, a position command $\mathbf{q}_{arm,d}$ is always translated into a trajectory following command by moving the hand along a trajectory $\mathbf{q}_{arm}(t)$, $t \in [0, T]$ such that: T is the execution time, $\mathbf{q}_{arm}(0)$ is the arm position when the command is received, $\mathbf{q}_{arm}(T) = \mathbf{q}_{arm,d}$ is the desired final position and finally $\dot{\mathbf{q}}_{arm}(t) = 0$, $\forall t > T$. If a velocity command $\dot{\mathbf{q}}_{arm,d}$ is received while executing a position command $\mathbf{q}_{arm}(t)$, the original velocity command is transformed into a new one: $\dot{\mathbf{q}}_{arm} = \dot{\mathbf{q}}_{arm}(t) + \dot{\mathbf{q}}_{arm,d}$.

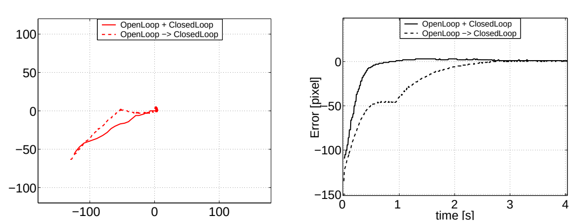

Fig. 7. Left: hand movement on the left image plane (320 × 240) during the execution of a single reaching movement. Dashed line: hand movement during an open loop movement followed by a closed loop phase. Solid line: hand movement during the superposition of open and closed loop strategies. Right: time response of the horizontal image plane position u_l. Dashed line: open loop movement followed by a closed loop phase. Solid line: superposition of open and closed loop.

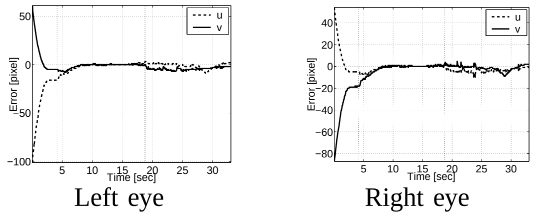

Fig. 8. Image plane movements during a three phase reaching. First the open loop, then the closed loop and finally a movement in the null space of the given task (keeping the hand in fixations). Each phase is delimited by a vertical dotted line.

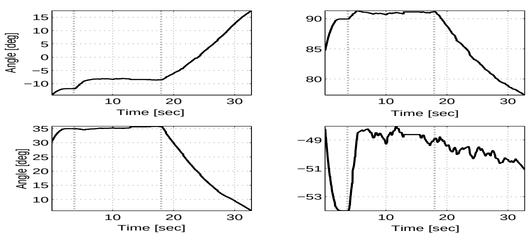

Fig. 9. Null space arm movement corresponding to the image plane movements shown in Figure 8.

task (keeping $\mathbf{u}_{hand} = 0$) as proposed in [10]. A simple way to perform this testing is the following control strategy:

$$\dot{\mathbf{q}}_{arm} = -k_1 \cdot \mathbf{J}^{\#}\mathbf{u}_{hand} + k_2(I - \mathbf{J}^{\#}\mathbf{J})\mathbf{w}, \qquad (12)$$

where $I \in \mathbb{R}^{4\times4}$ is the identity matrix, $\mathbf{w} \neq 0$ is a randomly chosen vector in $\mathbb{R}^4$ and k_1, k_2 are positive constants. Ideally, the strategy (12) should allow arm movements $\dot{\mathbf{q}}_{arm} \neq 0$ while leaving the hand position $\mathbf{u}_{hand}$ unperturbed. Practically we observed a minimal image plane movement (Figure 8) as oppposed to a relatively large arm movement (Figure 9). These results further prove the quality of our Jacobian estimation.

VI. Conclusions

We have described the implementation a 3D reaching controller. Our solution integrates an open loop and a closed loop phase. The former improves the action velocity allows to initiate a reaching movement when the hand is not visible. The latter employs visual feedback to improve precision.

The robot autonomously acquires the required sensory-motor transformations without prior information about the kinematic structure of the robot. The only simplification was the use of a color marker to visually localize the hand of the robot. Our assumption is that the hand localization/identification is a separate problem that is already solved when the robot starts learning to reach. (see for example in [17], [18]).

The proposed learning strategy allows autonomous estimation of the forward motor-motor map and of the eye-to-hand visual Jacobian. The estimation of the Jacobian is is fully autonomous and does not impose any constraints on the number of the degrees of freedom that are actuated.

Acknowledgement

This work is supported by the RobotCub project (IST-2004-004370), funded by the European Commission through the Unit E5 "Cognitive Systems". Moreover, it has been partially supported by Neurobotics, a European FP6 project (IST-2003-511492) and Contact (NEST 5010).

References

[1] J. M. Hollerbach and C. W. Wampler, "The calibration index and taxonomy for robot kinematic calibration methods," *International Journal of Robotics Research*, vol. 15, no. 6, pp. 573–591, 1996.

[2] R. Tsai and R. Lenz, "Real time versatile robotics hand/eye calibration using 3D machine vision," in *International conference on Robotics and Automation*, vol. 1, April 1988.

[3] N. Andreff, R. P. Horaud, and B. Espiau, "Robot hand-eye calibration using structure from motion," *International Journal of Robotics Research*, vol. 20, no. 3, pp. 228–248, Mar 2001.

[4] D. Bennett and J. Hollerbach, "Autonomous calibration of single-loop closed kinematic chainsformed by manipulators with passive endpoint constraints," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 5, pp. 597–606, October 1991.

[5] M. R. Blackburn and H. G. Nguyen, "Learning in robot vision directed reaching: A comparison of methods," in *ARPA Image Understanding Workshop*, Monterey, CA, USA, November 1994.

[6] G. Metta, G. Sandini, and J. Konczak, "A developmental approach to visually-guided reaching in artificial systems," *Neural Networks*, vol. 12, pp. 1413–1427, 1999.

[7] M. Lopes and J. Santos-Victor, "Learning sensory-motor maps for redundant robots," in *IEEE/RSJ International Conference for Intelligent Robots and Systems*, Beijin, China, October 2006.

[8] S. A. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996. [Online]. Available: citeseer.ist.psu.edu/hutchinson96tutorial.html

[9] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true Jacobian," in *Proc. IROS*, Sep. 1994.

[10] N. Mansard, M. Lopes, J. Santos-Victor, and F. Chaumette, "Jacobian learning methods for tasks sequencing in visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'06*, Beijing, China, October 2006, pp. 4284–4290.

[11] J. T. Laprest, F. Jurie, M. Dhome, and F. Chaumette, "An efficient method to compute the inverse jacobian matrix in visual servoing," in *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, vol. 1, New Orleans, LA, April 2004, pp. 727–732.

[12] B. Scassellati and G. Sun, "A fast and efficient model for learning to reach," *International Journal of Humanoid Robotics*, vol. to appear, 2007.

[13] C. Gaskett, A. Ude, and G. Cheng, "Hand-eye coordination through endpoint closed-loop and learned endpoint open-loop visual servo control," *International Journal on Humanoid Robotics*, vol. 12, no. 7, pp. 868–871, 1977.

[14] W. H. Press, *Numerical Recipes in C*. Cambridge: Cambridge University Press, 1990.

[15] S. Schaal and C. Atkenson, "Constructive incremental learning from only local information," *Neural Computation*, vol. 8, no. 10, pp. 2047–2084, 1998.

[16] C. Samson, B. Espiau, and M. L. Borgne, *Robot Control: the Task Function Approach*. Oxford University Press, 1991.

[17] L. Natale, F. Orabona, G. Metta, and G. Sandini, "Exploring the world through grasping: a developmental approach," in *6th CIRA Symposium*, Espoo, Finland, June 2005.

[18] A. Edsinger and C. Kemp, "What can i control? a framework for robot self-discovery," in *6th International Conference on Epigenetic Robotics*, Paris, France, September 2006.