Learning Sensory-Motor Maps for Redundant Robots

Manuel Lopes   José Santos-Victor

Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal

http://vislab.isr.ist.utl.pt

{macl,jasv}@isr.ist.utl.pt

Abstract— **Humanoid robots are routinely engaged in tasks requiring the coordination between multiple degrees of freedom and sensory inputs, often achieved through the use of sensory-motor maps (SMMs).**

Most of the times, humanoid robots have more degrees of freedom (DOFs) available than those necessary to solve specific tasks. Notwithstanding, the majority of approaches for learning these SMMs do not take that into account. At most, the redundant degrees of freedom (degrees of redundancy, DOR) are "frozen" with some auxiliary criteria or heuristic rule.

We present a solution to the problem of learning the forward/backward model, when the map is not injective, as in redundant robots. We propose the use of a "Minimum order SMM" that takes the desired image configuration and the DORs as input variables, while the non-redundant *DOFs* are viewed as outputs. Since the DORs are not frozen in this process, they can be used to solve additional tasks or criteria. This method provides a global solution for positioning a robot in the workspace, without the need to move in an incremental way. We provide examples where these tasks correspond to optimization criteria that can be solved online.

We show how to learn the "Minimum Order SMM" using a local statistical learning method. Extensive experimental results with a humanoid robot are discussed to validate the approach, showing how to learn the *Minimum Order SMM* of a redundant system and using the redundancy to accomplish auxiliary tasks.

I. INTRODUCTION

Humanoid robots must routinely coordinate the head and the arm. For this purpose, the robot must have a way to predict what will happen in the world if some action is made (forward model), and what action can change the world in a pre-defined manner (backward model). Usually, the correspondence between perception and action is called a *Sensory-Motor Map* (SMM) and it can be interpreted in terms of forward/inverse kinematics of robotic manipulators jointly with a camera. In this work, the SMM is used to predict the image resulting from the robot moving the arm to a certain posture, or the inverse association, by determining which motor command causes the arm to reach a specified appearance.

Quite often, humanoid robots have more degrees of freedom than those strictly necessary to accomplish a certain task. For example, Figure 1 shows several positions of an humanoid robot, where the wrist position is always the same, but the posture of the arm changes. In terms of input-output map, this redundancy translates into the fact that several different inputs yield the same observation or output. If the backward model is obtained by inverting the forward model, this causes a problem, because the function is no longer invertible. Also, if it is desired to learn the inverse model common algorithms will fail because the dataset is incoherent. Hence, to learn this map for a redundant systems requires the adoption of some extra assumptions. However, this strategy effectively "freezes" the redundant degrees of freedom, that can no longer be used for any additional task.
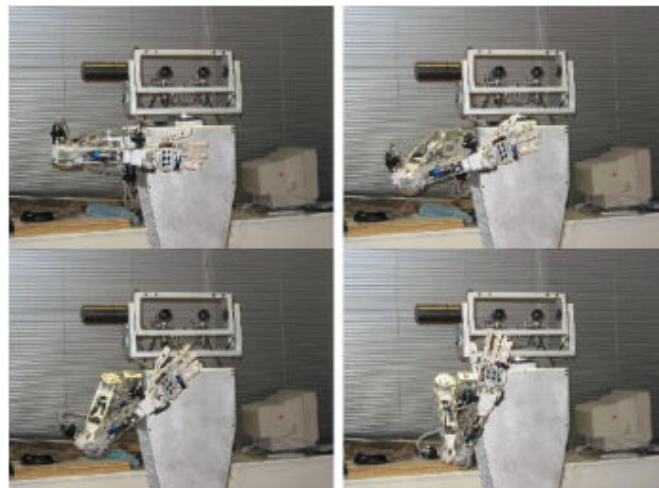

Fig. 1. Redundancy of the robotic system, the $3D$ position of the wrist is the same but the arm configuration is different

This work presents a solution for learning perception-action maps when redundant degrees of freedom exist (also known as degrees of redundancy, DOR). As we do not restrict the output of the system at learning time, the extra degrees of freedom stay free for online selection, and so they can be used to fulfill a secondary task or to meet an additional criterion or constraint, like e.g energy minimization. For this we propose the use of a "Reduced order SMM" that takes the image configuration and DORs as input variables, and the non-redundant DOFs as outputs.

We also present a methodology for estimating the "Reduced Order SMM" automatically. As the map should be differentiable, the function fitting method for estimating the map must be chosen carefully. We choose a local learning method with

differentiable kernels [1], [2].

Several applications have already been proposed, using the SMM perspective. A neural-network architecture was used in [3] to coordinate a binocular head with a three-DOF arm in a reaching task. As the open loop motion can have some error, this work also included some error correction methods by executing a closed loop visual servoing.

Correspondences between perception and action can also relate dynamic properties. In [4], a statistical learning method is used to learn an inverse kinematic function for an highly redundant humanoid robot, engaged in imitation tasks. In this case, a map from positions and velocities is related to image velocities, enabling the robot to repeat arm gestures. The trajectories for learning are hand-coded trajectories, the authors comment that this approach solves the problem of robot singularities, because only seen motions are learned, and so no infinite velocities appear.

The robot jacobian matrix [5] is very often used for control purposes. This matrix relates the cartesian velocity of the end-effector with the corresponding joint velocities. To move in a desired position the inverse jacobian should be evaluated. Some other methods can be used to do this inversion when the jacobian is not square [6]. A well known approach is the damped least-squares, where the inversion is made jointly with an energy minimization [7], originally introduced to solve the problem of controlling robots near singularities.

With redundant robots, the extra degrees of freedom can be used to solve other task, provided that the corresponding motion is done along a direction in the null space of the main motion, this is called the redundancy formalism as proposed in [8], [9]. Several criteria can be used to choose this secondary task. This formalism is very well described in [10] for humanoid animations. In humanoids, several conflicting constraints may frequently occur, like the position of the hands, feet and head in a dance posture. With the redundancy formalism these constraints can be dealt with. In a similar application to ours, the work of [11] presents a robot under visual control, where redundancy is used to obtain better trajectories in a visual servoing task.

We can find examples in the literature on work done in closed kinematic chains. Modeling the musculo-skeletal human system [12] requires some constraints, because the muscles only work by contraction, which must be taken into account when computing the solution. Other works have dealt with planning in humanoid robots. A system able to decompose the forces in order to act both the task domain, but also and independently in the robot posture is presented in [13].

Some methods have already been proposed to estimate the interaction or the jacobian matrix of robots. One of the first works was [14], where a very robust learning rule was derived and a convergence proof given. This method is based on the Broyden update rule already known from optimization theory [15]. It has been widely used in real robotic applications based on visual control. In [16], it was used for a grasping task guided by visual servoing. As the jacobian depends strongly on the current position, it must be evaluated at each time instant. For object-grasping task in [17] an estimation algorithm is used to provide an approximation of this highly non-linear mapping using several local linear models.

Our approach is different from these approaches in several ways:

- Includes visual information in the loop
- No knowledge about the system kinematics is needed. The sensory-motor map is learned with a self-exploration phase.
- The map is global and not a local approximation. This means that we can go directly to a position in an open-loop fashion if we want.
- Several criteria can be used as secondary task without having to learn a new map.

This paper is organized as follows: Section II describes the use of statistical learning methods in redundant robots. Section III shows how to use local regression methods to learn the partial backward model. Section IV is devoted to experiments done with a humanoid torso with 10 DOF, that evaluates the quality of our approach. Finally some conclusions and future work are done. As an annex, we have the deduction of the jacobian of the local learning method.

II. SENSORY-MOTOR COORDINATION WITH CONTROL OPTIMIZATION

In this section we show how to define a Sensory-Motor Map that explicitly takes the DOR into consideration, thus allowing the completion of several simultaneous tasks.

Let us define a *SMM* that maps a vector of control variables (n,r) to a vector of image point features $\mathcal{I}$, where n is a minimum set of degrees of freedom that spans the full output space and r is a set of redundant degrees of freedom. Note that there are several partitions of the input space, into redundant versus non-redundant degrees of freedom, that can give this same property. It is possible to find automatically the redundancy by analyzing the correlation matrix for the jacobian estimation [18]. This forward model can thus be written as:

$$\mathcal{I} = f(n,r)$$

and allows to predict the image configuration of the robot given a set of motor commands.

In many cases, we are more interested in the inverse map, i.e. computing the motor commands that drive the robot to a desired image configuration, $\mathcal{I}$. If there were an inverse mapping $(n,r) = f^{-1}(\mathcal{I})$, this problem could be solved in a straight forward manner. However, as the dimension of the input space is larger than that of the output space, there are many input combinations that generate the same image point features. In other words, because of the DOR, $f(n,r)$ is not bijective and, therefore, not invertible.

Fig. 1 shows an example of redundancy, where, for this robot the $3D$ position of the wrist is controlled with 4 DOFs, thus remaining one DOR.

To put the problem in another perspective, we can say that finding the robot joint angles to move the arm to a desired image configuration $\mathcal{I}$ is an ill-posed problem when the arm has redundant degrees of freedom, [19], because multiple solutions exist.

One approach to solve ill-posed problems, [20], [21], consists in using additional constraints that restrain the set of admissible solutions, in such a way that the solution sought becomes unique in this reduced solution space. In our case, this corresponds to recast the original problem to that of moving the robot to a desired image position $\mathcal{I}^*$ while, at the same time, minimizing some auxiliary criterion, $c(n, r)$.

We built a cost function, $\mathcal{K}$, with two terms: one weighting the error in the position of the end effector (data fitness) and another one corresponding to the weights on the control (regularization term).

$$\mathcal{K}(\mathcal{I}^*, n, r) = \lambda \|\mathcal{I} - \mathcal{I}^*\|^2 + c(n, r) \qquad (1)$$

This cost function expresses that we are willing to accept some error in the position if another task can be solved at the same time, in this case control costs. Examples of control cost criteria c can be "Comfort" (e.g. distance to joint limits), Energy minimization (e.g. the position with lower momentum) or Minimum motion (i.e. minimize total motion from current to desired position), posture control, amongst others.

The regularized solution can be found by minimizing the cost defined in Equation (1), as follows:

$$(\hat{n}, \hat{r}) = \arg \min_{n, r} \left(\lambda \|\mathcal{I} - \mathcal{I}^*\|^2 + c(n, r) \right) \qquad (2)$$

where $\mathcal{I}$ can be computed with the forward model $\mathcal{I} = f(n, r)$. Similarly to [13], this formula integrates two terms: one describing the task part and another related to posture control.

There are two important observations to this formulation. Firstly, the optimization is done with respect to all control variables, which translates into a significant computational cost. Secondly, the DORs are not treated as such, since they undergo exactly the same process as the non-redundant DOFs.

The consequence of this approach is that the extra degrees of freedom are frozen from the beginning and can no longer be used for a different purpose during execution. In a way, redundancy is lost.

Instead, in our approach, we would like to keep the redundant degrees of freedom free for solving additional tasks online. In essence, we split the problem in two steps. Firstly, we define a "Minimal Order Sensory Motor Map", $g(\mathcal{I}, r)$, that relates n and $(\mathcal{I}, r)$:

$$n = g(\mathcal{I}, r) \qquad (3)$$

By taking the DORs as input (independent variables) instead of output signals, the problem of computing the non-redundant DOFs becomes well posed. The DORs, r, are left unconstrained and can be fixed during runtime, when a secondary task or optimization criterion is specified.

The definition of the "Minimum Order SMM" allows us to use the redundancy to meet additional criteria or task-constraints, that can be changed online. The DORs can be determined as the solution of a new optimization problem, with cost function $\mathcal{L}$:

$$\hat{r} = \arg \min_r \mathcal{L}(\mathcal{I}^*, r) \qquad (4)$$

The optimization is done with a gradient-descendant method with following update step:

$$r_{t+1} = r_t - \alpha \nabla_r \mathcal{L}(\mathcal{I}, r)$$

Note that, in contrast with the previous case, this optimization is done with respect to the redundant degrees of freedom, only. The optimization complexity is thus substantially lower and lends itself to be used as an online process. In general, the solutions in the two cases are not the same, because different local minima could be reached and the criteria are slightly different.

Our approach guarantees zero prediction error, because the *Minimum Order SMM* allows us to determine the values of n corresponding to the exact image position, for the selected redundant degrees of freedom. This solution is similar to the first (regularized) problem when λ becomes large. If the *Minimum Order SMM* is not exact, then it will introduce some error in the final image configuration.

For clarity, we summarize the final algorithm.

1) Select the desired image configuration, $\mathcal{I}^*$
2) Select and initial motor command (n, r)
3) Select the secondary task optimization criterion
4) Solve the optimization of Equation (4) for r and use $g(.)$ to compute n.
5) Move the arm to the obtained solution, (n, r)
6) Observe $\mathcal{I}$ and possibly adjust the function $g(\mathcal{I}, n)$
7) If some extra precision is needed, go to 4

There are several important differences in our approach whem compared to other methods based on the robot Jacobian. The *Minimum Order SMM* provides directly the goal position corresponding to the desired redundant joint position. It is then possible to move the robot directly (i.e. in an open-loop fashion) to the goal position, avoiding incremental steps. The posture optimization is done iteratively with the previous update rule. Therefore, the motion goes along the optimization path or directly to the convergence point. This is the case because no visual feedback is necessary to the algorithm. If extra precision is needed, then a visual feedback loop needs to be added.

An example, where the secondary goal is to maintain the control variables as near zero as possible, is presented next:

$$\begin{aligned} \mathcal{L}(\mathcal{I}^*, r) &= \|n\|^2 + \|r\|^2 \\ &= \|g(\mathcal{I}^*, r)\|^2 + \|r\|^2 \qquad (5) \end{aligned}$$

Differentiating this cost function yields:

$$\nabla_r \mathcal{L}(\mathcal{I}, r) = 2 \left(\frac{\partial g(\mathcal{I}, r)}{\partial r} g(\mathcal{I}, r) + r \right)$$

The derivation of $\frac{\partial g(\mathcal{I}, r)}{\partial r}$ is presented as an appendix.

We have seen how the introduction of the *Minimum Order SMM* allows us to use the system redundancy to solve additional tasks online, as opposed to freezing the DORs in a regularized solution to the initial ill-posed problem. In the next section, we will see how to estimate the *Minimum Order SMM* $g(\mathcal{I}, n)$ online.

III. Learning the Minimum Order SMM through Local Regression

In the previous section we have seen how to partition the redundant and non-redundant degrees of freedom to build a Minimum Order SMM, $g(\mathcal{I}, r)$ that allows for the computation of the non-redundant DOFs leaving the DORs unconstrained. We will now see how such a map can be estimated online. Without loss of generality, let us assume that we want to estimate the following non-linear function:

$$y = f(x) \tag{6}$$

Since we have little information about this function, the usual approach consists in approximating $f(x)$ by a set of models that are good local approximations of the original global non-linear function, [1].

In this work, $f(x)$ will be approximated by a mixture of models that are locally linear. Obviously, a single linear approximation would fail to provide the desired degree of accuracy. Each local model has a "confidence" region, the kernel K_j, the mixture of all models yields the approximation:

$$y = f(x) \approx \frac{\sum_{j=1}^{M} K_j B_j^T x}{\sum_{j=1}^{M} K_j}$$

for some regression matrices, B_j to be estimated. The choice of the kernel shapes [2] leads to different properties of the approximating function. We have adopted a Gaussian kernel with mean μ and variance W:

$$K_j = K_{W_j}(\mu_j, x) = \frac{1}{det(W_j)} e^{-(x-\mu_j)^T W_j (x-\mu_j)} \tag{7}$$

Let us assume for the moment that the number and the parameters of each Kernel are known in advance. Each model will be fitted by minimizing the following criteria:

$$\hat{B}_j = \arg\min_B \sum_{i=1}^{t} \lambda^{(t-i)} K_j \left\| y_i - B^T x_i \right\|^2 \tag{8}$$

where K_j weights points according to the kernel measure and λ provides a time forgetting factor. The model can be estimated by:

$$\hat{B} = Q R^+ \tag{9}$$

with

$$Q = \sum_{i=1}^{t} \lambda^{(t-i)} K_W(\mu, x_i) y_i^T x_i$$

$$R = \sum_{i=1}^{t} \lambda^{(t-i)} K_W(\mu, x_i) x_i^T x_i \tag{10}$$

An advantage of writing these terms in this way is the possibility of defining an online estimator:

$$Q_t = \lambda Q_{t-1} + K_W(\mu, x_t) y_t^T x_t$$
$$R_t = \lambda R_{t-1} + K_W(\mu, x_t) x_t^T x_t \tag{11}$$

Finally, at runtime when an input sample is present, the output will be evaluated as a combination of each model B_i weighted by K_j:

$$\hat{y} = \frac{\sum_{j=1}^{M} K_j \hat{B}_j^T x}{\sum_{j=1}^{M} K_j} \tag{12}$$

If the kernels are C^0 and have an infinite support, this function is guaranteed to be C^0. If the kernels are differentiable, the function will be C^1.

The final point to discuss is related to the Kernel functions $K_W(\mu, x)$. How many kernels should be used and what should the parameters of each kernel be? The number of kernels can be is iteratively increased during training. When the distance between a new data sample and its nearest kernel exceeds a certain threshold, a new kernel is created with center (μ) in this point. The shape of the kernels (the covariance matrix) can be automatically updated choosing, e.g., a measure of reconstruction quality [22], [23].

Other formulations have already been proposed. The *Locally Weighted Projection Regression* method, proposed in [24], is linear with the number of samples and every new sample can be added easily. As the method is not capable of extrapolating, the work space must be well covered in the training set. Other implementations keep several samples in memory without estimating any explicit models. The prediction is produced online by weighting the points in memory with some kernel functions.

IV. Experimental Results

Several experiments were done with a real robot to assess the quality of the algorithms and the ability to learn the proposed SMM online. The experimental setup was the Baltazar humanoid robot torso [25], consisting of a 4 DOF head, a 6 DOF arm and a 10 DOF under-actuated hand. The image features consist on the image position of the wrist, and we want to position the wrist in the the image. This task requires only two degrees of freedom. The non-redundant DOFs are the shoulder adduction/abduction and flexion/extension. The shoulder axis rotation and elbow flexion/extension are considered as redundant degrees of freedom.

A. Evaluation of the learning method

The first experiment is designed to validate the learning of the Minimum order SMM. The map associates the image position of the robot hand and the DOR to the non-redundant DOFs.

During learning, the head remains in a fixed position and observes the robot hand. The arm is moved to several randomly selected positions. The range of movement of each joint is in the order of $0.55\,rad$. Once, the robot wrist attains one

such position, it remains fixed while different solution for the inverse kinematics are used. This is possible due to the redundancy of this robot's kinematics, when four degrees of freedom are used for a positioning task. Another redundancy exists because, with the use of a single camera, depth information is lost. Hence, for a 2 DOF task we have 4 DOF available.

Fig. 2 shows the evolution of joint angles during this period of auto-observation and learning. It also shows the image position of the robot wrist where, due to elasticity in the robot joints, oscillations occur when the acceleration is high.
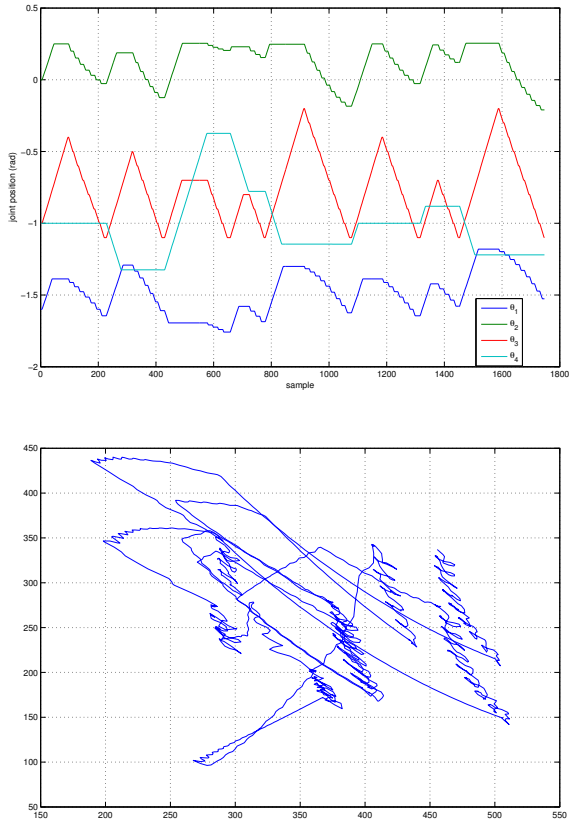

Fig. 2. Dataset for the real robot experiments. The top figure shows the temporal evolution of the joint angles (joint position in radian vs sample number), the bottom presents the trajectories in the image (pixel coordinates), where oscillation is caused by elasticity in the robot joints.

Fig. 3 shows the quality of the SMM estimation, as described in Section III. The top plots show the true and estimated non-redundant joint angles, which are in good agreement. The histogram and cumulative distribution of the error are shown in the bottom plots, for 2000 data points. For both non-redundant joints less than 10% of the points have an error bigger than $0.05\,rad$.

These results show that the online estimation method presented in Section III provides a good approximation to the original Minimum order SMM. The next set of experiments show how to define a secondary task based on an energy minimization criterion, to drive the robot to the desired position, while meeting this secondary goal.
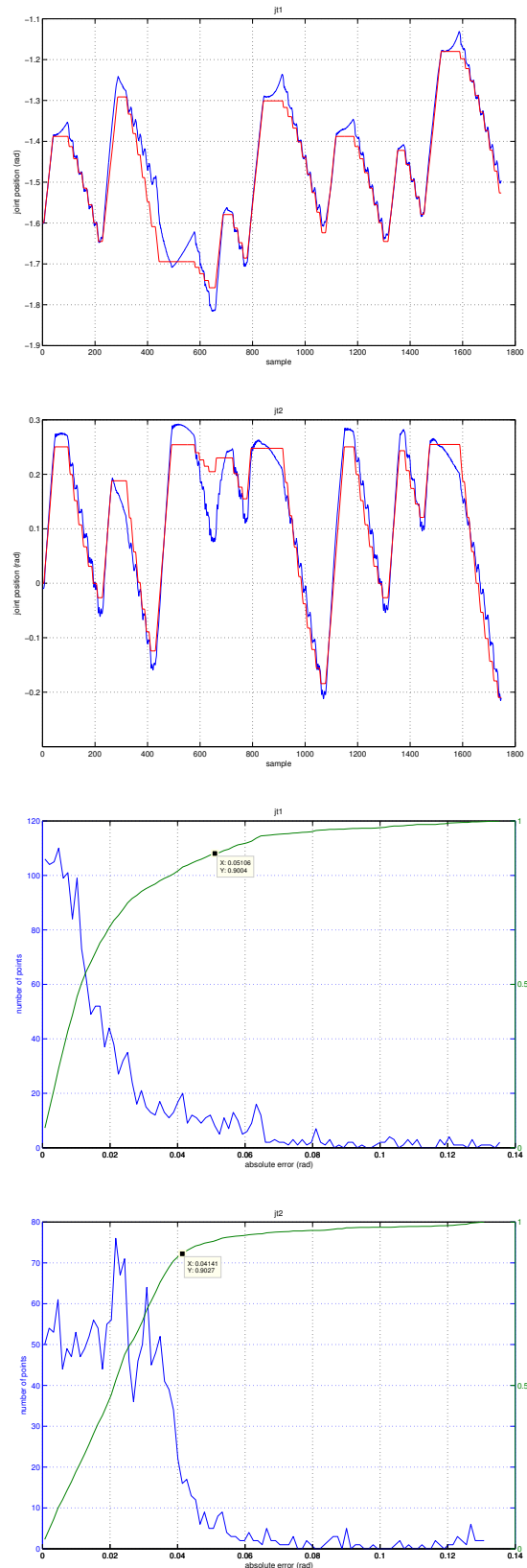

Fig. 3. These figures show the prediction error for (non-redundant) Joint 1 and Joint 2. The absolute error histogram and the cumulative distribution are also shown. Almost 90% of the samples have an error below $0.05\,rad$, see measure in the figures.

B. Sensory-motor coordination for redundant robots

For a given desired image position and an initial position of the redundant degrees of freedom, our goal is to reach a certain image position, while satisfying a secondary criterion (task). This is obtained through the following optimization problem, as defined earlier:

$$\mathcal{L} = \|n - \mu_n\|^2 + \|r - \mu_r\|^2$$

that aims to maximize the distance to joint limits, corresponding to a comfort criterion.

It is worth stressing that the optimization process relies on the estimated Minimum order SMM, as described before. Fig. 4 presents the evolution of the cost function l, for each iteration of the Newton method. It also presents the trajectory of all 4 (redundant and non-redundant) robot joints. We can see that, for this case, the maximum for one joint was $0.5\,rad$.
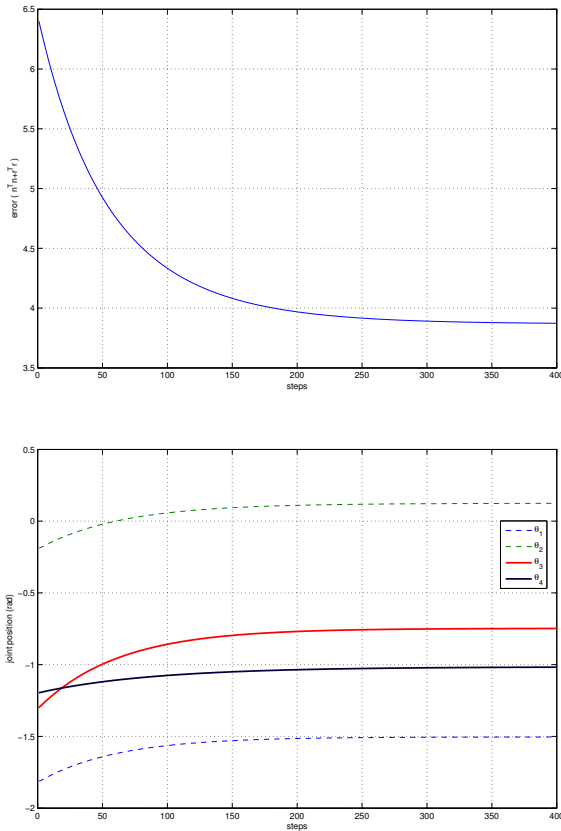

Fig. 4. Convergence rate and evolution of the position for the real robot as a function of the optimization step. It is interesting to see that one joint moved $0.5\,rad$ and the final error in the image corresponds to $0.03\,rad$.

The final error in the image was as small as $0.03\,rad$, and most it is due to elasticity in the robot joints. Fig. 5 shows the robot view of the hand for an intuition for this error (about the size of the target). Due to the redundancy in the arm, it would be possible to fixate the target while changing the arm posture.
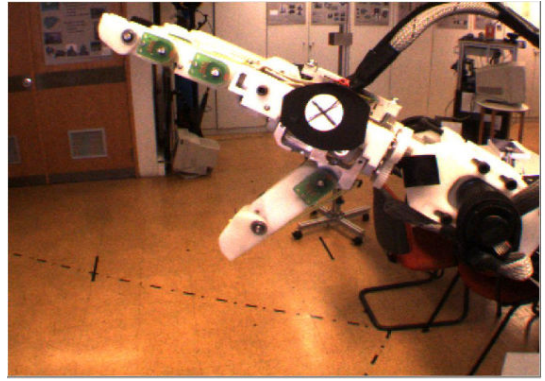

Fig. 5. Robot view. We can see the arm, hand and the target being tracked

V. CONCLUSIONS / FUTURE WORK

We have addressed the problem of estimating Sensory Motor Maps in redundant systems, which is often the case of humanoid robots. As a consequence of redundancy, the inverse map cannot be estimated since the forward model is not bijective.

For a given task, we started by partitioning the robot degrees of freedom in redundant and non-redundant DOFs. Then, we defined a "Minimum order SMM" that takes as the input general image configurations and redundant degrees of freedom. This partial backward model can be used to determine the configuration for the non-redundant degrees of freedom and can be used for control. Using the "Minimum order SMM", the redundant degrees of freedom are available to meet additional online constraints, arising from secondary tasks or criteria.

A noteworthy observation is that this method is not incremental, in the sense of requiring small steps toward the final goal. It gives directly the goal position so that the robot can be moved directly there. However, the optimization leads to several steps being used for the secondary goal, in this case the posture optimization.

The Minimum Order SMM is learned with a local learning method. Experimental results done in an humanoid torso with 10 degrees of freedom were presented, illustrating both the ability to learn the *Minimum order SMM* and how it can be used for specific tasks.

A large workspace was used of about $40\,degrees$ for each joint, a small reconstruction error was achieved, about $2.5\,degrees$. For the optimization one joint could "travel" $90\,degrees$ to be able to reduce the cost function by 40%, with a corresponding error in the image of only $1.5\,degrees$ (about the same size of the target).

In the future we plan to investigate automatic methods for the division between redundant and non-redundant degrees of freedom. One possible direction of research can go a similar path as the one presented in [18]. The use of more degrees of freedom will be necessary to deal with more complex image features (e.g. position and orientation), a binocular head and changing head positions.

As a final comment, we would like to stress that learning

inverse maps for redundant robots is a frequent need in humanoid robotics, and that the proposed method is simple, computationally efficient and well suited for online learning.

APPENDIX

A. Derivative of Local learning method

If redundant degrees of freedom are to be chosen with some extra criterion, it is important to evaluate the derivative of the prediction function. The prediction of the chosen local learning method is given by:

$$\hat{y} = \frac{\sum_{j=1}^{M} K_j B_j^T x}{\sum_{j=1}^{M} K_j}$$

Now we want to evaluate its derivative as a function of the inputs $\frac{\partial \hat{y}}{\partial x}$:

$$\frac{\partial \hat{y}}{\partial x} = \sum_{j=1}^{M} \left(\frac{\partial K_j}{\partial x} x^T + K_j \right) \frac{B_j}{\tilde{k}} - \frac{1}{\tilde{k}^2} \sum_{l=1}^{M} \frac{\partial K_l}{\partial x} \sum_{j=1}^{M} K_j x^T \beta$$

with $\tilde{k} = \sum_{j=1}^{M} K_j$ and $\frac{dK_j}{dx} = -W_j(x - \mu_j)K_j$

After some computations, we have:

$$\tilde{k} \frac{\partial \hat{y}}{\partial x} = \sum_{j=1}^{M} \frac{\partial K_j}{\partial x} x^T B_j + \sum_{j=1}^{M} B_j K_j - \tilde{dk} \, \hat{y}^T \qquad (13)$$

with $\tilde{dk} = \sum_{j=1}^{M} \frac{\partial K_j}{\partial x}$

ACKNOWLEDGEMENTS

REFERENCES

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[2] Chistopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.

[3] M. Blackburn and H. Nguyen. Learning in robot vision directed reaching: A comparison of methods. In *ARPA Image Understanding Workshop*, Moterey, CA, 1994.

[4] Aaron D'Souza, Sethu Vijayakumar, and Stephan Schaal. Learning inverse kinematics. In *International Conference on Intelligent Robots and Systems*, Hawaii, USA, 2001.

[5] John J. Craig. *Introduction to Robotics*. Addison-Wesley Pub Co, 1989.

[6] Samuel R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Technical report, University of California, San Diego, USA, 2004.

[7] C.W. Wampler. Manipulator inverse kinematics solution based on damped least-squares solutions. *IEEE Trans. Systems, Man and Cybernetics*, 16(1), 1986.

[8] J.B. Rosen. The gradient projection method for nonlinear programmimg, part i, linear constraints. *SIAM Journal of Applied Mathematics*, 8:181–217, 1960.

[9] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.

[10] Paolo Baerlocher and Ronan Boulic. An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *Visual Computer*, 2004.

[11] Nicolas Mansard and François Chaumette. Tasks sequencing for visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, September 2004.

[12] Yoshihiko Nakamura, Katsu Yamane, Ichiro Suzuki, and Yusuke Fujita. Dynamics computation of musculo-skeletal human model based on efficient algorithm for closed kinematic chains. In *2nd International Symposium on Adaptive Motion of Animals and Machines*, Kyoto, Japan, 2003.

[13] Oussama Khatib, Oliver Brock, Kyong-Sok Chang, Diego Ruspini, Luis Sentis, and Sriram Viji. Human-centered robotics and interactive haptic simulation. *International Journal of Robotics Research*, 23(2), 2004.

[14] Koh Hosoda and Minoru Asada. Versatile visual servoing without knowledge of true jacobian. In *International Conference on Intelligent Robots and Systems*, pages 186–193, Munchen, Germany, Sep 1994.

[15] R. Fletcher. *Practical Methods of Optimization*. Chichester, 2nd edition, 1987.

[16] M. Jagersand and R.C. Nelson. On-line estimation of visual-motor models using active vision. In *ARPA Image Understanding Workshop*, pages 677–682, 1996.

[17] Manuel Lopes, Alexandre Bernardino, and José Santos-Victor. A developmental roadmap for task learning by imitation in humanoid robots. In Yannis Demiris, editor, *AISB - Third International Symposium on Imitation in Animals and Artifacts*, Hatfield, Uk, April 2005.

[18] Koh Hosoda and Minory Asada. How does a robot find redundancy by itself? In *Advances in Robot Learning: 8th European Workshop on Learning Robots*, Lausanne, Switzerland, 1999.

[19] A.N. Tikhonov and V.A. Arsenin. *Solution of ill-posed problems*. Washington DC: Winston, 1977.

[20] T. Poggio, V. Torre, and C. Kock. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[21] M. Bertero, T. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, 1988.

[22] S. Schaal and C. G. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084, 1998.

[23] Trevor Hastie and Clive Loader. Local regression: Automatic kernel carpentry. *Statistical Science*, 8:120–129, 1993.

[24] S. Vijayakumar and S. Schaal. Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional spaces. In *ICML*, Stanford, USA, 2000.

[25] Manuel Lopes, Ricardo Beira, Miguel Praça, and José Santos-Victor. An anthropomorphic robot torso for imitation: design and experiments. In *IEEE/RSJ - International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.