

Learning an internal representation of the end-effector configuration space

Alban Laflaquière¹, Alexander V. Terekhov¹, Bruno Gas¹ and J. Kevin O'Regan²

Abstract—Current machine learning techniques proposed to automatically discover a robot's kinematics usually rely on *a priori* information about the robot's structure, sensor properties or end-effector position. This paper proposes a method to estimate a certain aspect of the forward kinematics model with no such information. An internal representation of the end-effector configuration is generated from unstructured proprioceptive and exteroceptive data flow under very limited assumptions. A mapping from the proprioceptive space to this representational space can then be used to control the robot.

I. INTRODUCTION

One of the problems an autonomous robot must be able to solve is to retrieve basic information about its own topological structure relying on minimal *a priori* information. The problem of learning the kinematics of a manipulator has been approached by a number of machine learning techniques (see review in [10]). It is usually assumed that the end-effector position is registered using an external measuring system and the problem consists in finding a mapping between the joint space and the end-effector position. Some attempts have been made to learn so-called 'visual-motor' mappings between the joint angles and end-effector image in the robot-mounted visual system [2]. These approaches rely heavily on *a priori* assumptions about the properties of sensors and the robot structure. In alternative approaches no such assumptions are made, but they require the robot surface to be covered with an artificial skin [9], [5].

Imagine a robotic arm with an unspecified number of links and a camera installed on one of them. The camera output is scrambled, so that the order of pixels is completely broken and some of them are missing. The robot measures its configuration with uncalibrated encoders in the joints, but as the geometrical properties and link topology are unknown, these measurements do not tell anything about the actual configuration of the robot in space. Whereas such a robot does not have any explicit measure of its end-effector position, this information is implicitly available if the output of the camera is different for different positions / orientations. This implicit information can be sufficient to build an internal representation of the end-effector space, which has the same topology as the actual end-effector space but may have different metrics. Such a representation simply associates

together all configurations of the robot for which the camera position and orientation are the same, and thus introduces a new topology on the space of the encoder outputs. Building such a representation would be a trivial task if the camera position and orientation were available. In our case they are only accessible through the scrambled picture it provides. In this study we present an algorithm that builds an internal representation of the end-effector configuration space, from the unordered and scrambled information provided by the camera (or any other exteroceptor) installed on the end-effector.

The overall idea is illustrated in Fig. 1. The robot accesses its kinematic state through proprioception (e.g. encoders in joints) but has no information about its geometry or topological structure. The state of the environment is reflected in the outputs of the exteroceptors (camera, microphone array, set of photodiodes, etc). When the environment does not change, the output of the exteroceptors is determined by the position of the end-effector. The aim of the algorithm is to build an internal representation of the end-effector space by associating all proprioceptive inputs that correspond to the same outputs of the exteroceptor. Note that the topology of the internal representation can be different from the topology of the true end-effector space if the exteroceptors have the same outputs for different end-effector positions.

We present a neural network-based algorithm that maps a potentially high dimensional proprioception space into a low dimensional space of internal representation. The dimension of the latter is assumed known here and can be estimated using methods like those in [6], [3]. We test the quality of the mapping by making a simulated robot perform reaching movements using the Jacobian of the mapping defined by the neural network.

II. METHOD OVERVIEW

The method section consists of three parts. The first part describes a learning algorithm for the neural network (NN) mapping proprioception \mathcal{P} to an internal representation Ξ , which is topologically equivalent to the space of end-effector configurations \mathcal{C} . The Jacobian of this mapping is computed in the second part. In the third part, the routine used for the reaching movement is described.

A. Neural network and cost function

The goal of the NN is to map elements p of the proprioception space \mathcal{P} into the elements ξ of an *a priori* unspecified internal representation space Ξ . The only constraint on this space is that it must be topologically identical to the

* This work was conducted within the French/Japan BINAAHR (BINAural Active Audition for Humanoid Robots) project under Contract n° ANR-09-BLAN-0370-02 funded by the French National Research Agency.

¹ is with UPMC Univ Paris 06 and ISIR (CNRS UMR 7222), F-75005, Paris, FRANCE name@isir.upmc.fr

² is with Univ Paris 05 Descartes and LPP (CNRS UMR 8158), F-75006, Paris, FRANCE jkevin.oregan@gmail.com

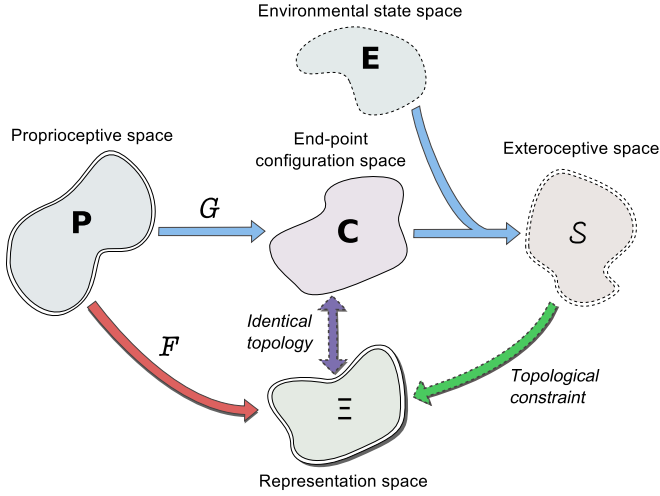


Fig. 1: Double-lined spaces are accessible to the robot. Dotted-lined spaces may change for different environments. Each proprioceptive state p generates an end-effector configuration c associated with an exteroceptive state s that depends on the environmental configuration. The goal of the robot is to create an internal representation ξ of the end-effector configuration using topological information of the exteroceptive space.

space of end-effector configurations \mathcal{C} . As the end-effector configuration c is unknown to the robot, the difference in the exteroceptors s is used instead to generate Ξ . This approach implies that the robot's sensory experience is not ambiguous, *i.e.* each exteroceptive state s is uniquely related to one configuration c of the end-effector. Each configuration c can however be related to multiple proprioceptive states p if the robot is redundant.

1) *Neural network architecture:* We used a multilayer perceptron (MLP [8]) with one hidden layer and a sigmoid excitation function σ . Each weight $w_{l,k}$ is initially drawn from a uniform distribution in $[-1, 1]$. Excitation y_k of neuron k is computed as follows:

$$y_k = \sigma\left(\sum_l w_{l,k} y_l\right) = \sigma(e_k) \quad (1)$$

with y_l the excitation of neuron l in the previous layer and $\sigma(\cdot)$ the sigmoid function.

The number of input neurons N^{in} is equal to the dimension of the proprioceptive space \mathcal{P} . The number of hidden neurons N^h is arbitrarily set to 30. It could naturally be modified according to the expected mapping complexity. The number of output neurons N^{out} is equal to the number of independent variables necessary to describe the exteroceptive state s . In the following, N^{out} is assumed known, but its value could be determined through an exteroceptive manifold analysis [6], [3]. Note that if the robot's sensory capacity does not allow a complete characterization of the end-effector configuration c , the number of independent variables N^{out} can be smaller than expected from an external point of view.

2) *Cost function minimization:* As it is difficult to build a cost function that only captures the similarity of the topology,

we required the NN to preserve in Ξ the distances between the points in the exteroception space \mathcal{S} . The cost function is:

$$Q = \sum_{i,j} Q^{i,j},$$

$$Q^{i,j} = h\left(\|s^{i,j}\|\right) \left(\|\xi^{i,j}\| - \|s^{i,j}\|\right)^2 + \gamma \left(\|\xi^i\|^2 + \|\xi^j\|^2\right)$$

with:

$$\|s^{i,j}\| = \|s^i - s^j\| \text{ and } \|\xi^{i,j}\| = \|\xi^i - \xi^j\| \quad (2)$$

where $\|\cdot\|$ denotes the euclidean norm, ξ^i (resp. ξ^j) is the network output for the proprioceptive configuration p^i (resp. p^j), s^i (resp. s^j) is the exteroceptive state associated with p^i (resp. p^j), γ is a small centering factor added for the purpose of regularization, and $h(\cdot)$ is a neighborhood function designed to give more weight to small exteroceptive distances. In the following, $h(\cdot)$ is a simple step function:

$$h(a) = \begin{cases} 1 & \text{if } a \leq \mu \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Unlike traditional MLP cost functions, $Q^{i,j}$ is defined for pairs of inputs $\{p^i, p^j\}$ and outputs $\{\xi^i, \xi^j\}$. Only distances $\|\xi^{i,j}\|$ in the output space Ξ are then constrained. Moreover, this constraint is defined through auxiliary distances $\|s^{i,j}\|$. The first term of the sum is minimized when outputs distances $\|\xi^{i,j}\|$ tend to be equal to exteroceptive distances $\|s^{i,j}\|$. The second term is a centering cost that is minimized when outputs ξ tend towards 0 (Note that γ must be small enough not to disturb the first term minimization). Conservation of the exteroceptive space topology is achieved by respecting only small exteroceptive distances $\|s^{i,j}\|$ through the use of the neighborhood function $h(\cdot)$.

We used RPROP algorithm to find a minimum of the cost function [7]. The algorithm updates the weights $w_{l,k}$ according to the sign of the local gradient $\frac{\partial Q^{i,j}}{\partial w_{l,k}}$ defined as follows:

- if neuron k is in the output layer

$$\frac{\partial Q^{i,j}}{\partial w_{l,k}} = h(\|s^{i,j}\|) \left(\delta_k^i y_l^i + \delta_k^j y_l^j \right) \quad (4)$$

$$\delta_k^i = 2 \left(\gamma (y_k^i + y_k^j) + (y_k^i - y_k^j) \left(1 - \frac{\|s^{i,j}\|}{\|\xi^{i,j}\|} \right) \right) \sigma'(e_k^i)$$

$$\delta_k^j = 2 \left(\gamma (y_k^i + y_k^j) - (y_k^i - y_k^j) \left(1 - \frac{\|s^{i,j}\|}{\|\xi^{i,j}\|} \right) \right) \sigma'(e_k^j)$$

- if neuron k is in the hidden layer

$$\frac{\partial Q^{i,j}}{\partial w_{l,k}} = y_l^i \sigma'(e_k^i) \sum_n \delta_n^i w_{k,n} + y_l^j \sigma'(e_k^j) \sum_n \delta_n^j w_{k,n} \quad (5)$$

where $\sigma'(\cdot)$ is the derivative of the excitation function (sigmoid).

3) *Exploration and learning:* The data necessary to perform the learning are generated through a random exploration of the robot's working space. First, N proprioceptive states p^i are randomly selected in the neighborhood of the reference configuration. The neighborhood was defined so that the deviation of each proprioceptive value p_k^i did not exceed a predefined value A . The configurations falling out of the

working space were discarded. Then the exteroceptive values s^i associated with those proprioceptive states are computed. The proprioceptive commands p^i and exteroceptive states s^i are normalized to avoid saturation of the sigmoid excitation function:

$$\tilde{p}^i = 0.8(p^i - \bar{p}) / \max_{i,k}(|p_k^i - \bar{p}_k|) \quad (6)$$

$$\tilde{s}^i = 0.4(s^i - \bar{s}) / \max_{i,k}(|s_k^i - \bar{s}_k|) \quad (7)$$

Finally, all possible pairs $\{\tilde{p}^i, \tilde{p}^j\}$ with $j \in \{i+1, N\}$, and corresponding $\{\tilde{s}^i, \tilde{s}^j\}$, are created to form the learning base. The RPROP algorithm is applied for T iterations or until the error Q reaches a minimal value Q_{min} .

In most cases the cost function (2) has numerous local minima, that make its minimization difficult. In order to assure better initial conditions we performed a specific initialization of the network. We generated an N^{out} -dimensional unfolding of the exteroceptive set $\{\tilde{s}^i\}$ using Isomap [11]. The usual quadratic error minimization was then performed using the resulting projection as the desired values ξ of the NN [8].

This preliminary learning is supposed to bring the network into a favourable basin of attraction before the less-constrained cost minimization described in II-A.2 is applied. Note that the NN initialization is performed for a fixed environment. The latter can however be different and even change during the Q minimization as long as distances $\|s^i - s^j\|$ are computed for a single environment (see (2)). In the following, three random environments are considered, leading to three successive explorations during the minimization of Q .

B. Jacobian estimation

Once trained, the NN defines a mapping F from the proprioceptive space \mathcal{P} to the internal representation space Ξ , topologically equivalent to the configuration space \mathcal{C} of the end-effector (see Fig.1).

The mappings F and the corresponding spaces Ξ vary from run to run. What should remain invariant however, is that if any two configurations of the robot p^1 and p^2 are mapped to the same ξ by the function F computed in the first run, then these configurations must be also mapped to the same value ξ' by the function F' computed in any other trial. Moreover, if two configurations c^1 and c^2 are close in \mathcal{C} , their corresponding representations ξ^1 and ξ^2 must be close in the internal space Ξ .

This simply implies that the Jacobians of the functions F and F' have the same null-spaces (kernel) at every point p and their null-spaces coincide with the null-space of the mapping

$$c = G(p)$$

relating the configuration of the robot to the real end-effector position.

The Jacobian of F is defined as

$$J_F(p) = \frac{\partial F}{\partial p} = \begin{pmatrix} \frac{\partial \xi_1}{\partial p_1} & \cdots & \frac{\partial \xi_1}{\partial p_{Np}} \\ \vdots & \vdots & \vdots \\ \frac{\partial \xi_n}{\partial p_1} & \cdots & \frac{\partial \xi_n}{\partial p_{Np}} \end{pmatrix}. \quad (8)$$

The partial derivatives $\partial \xi / \partial p_k$ were computed using Euler's formula

$$\frac{\partial \xi}{\partial p_k} = \frac{1}{\epsilon} \left(F(p + \epsilon p_k) - F(p) \right) \quad (9)$$

where p_k is the k -th basis vector of the proprioceptive space and ϵ is a small value.

For later comparison, the Jacobian J_G of G is determined analytically.

C. Use of the estimated Jacobian in a reaching task

The trained NN captures the topological properties of the robot's forward kinematics and can then be used to perform a reaching task. Let $p(0)$ be an initial proprioceptive configuration in the portion of the proprioception space \mathcal{P} explored during the learning phase and $\xi(0)$ be the output associated with $p(0)$. Let ξ^* be the desired configuration in the internal representation space Ξ . The Jacobian J_F of F can be used to determine the shortest trajectory in the proprioceptive space such that internal state $\xi(t)$ moves along a straight line from $\xi(0)$ to ξ^* . The reaching algorithm is as follows, with t the current iteration:

- estimate the Jacobian in $p(t)$ (see (8), (9)).
- compute the proprioceptive configuration update:

$$\Delta p(t) = J_F^+ \Delta \xi = J_F^+ (\xi^* - \xi(t)), \quad (10)$$

where J_F^+ denotes the Moore-Penrose pseudoinverse of J_F .

- update proprioceptive configuration with a fixed step size ϵ :

$$p(t+1) = p(t) + \epsilon \frac{\Delta p(t)}{\|\Delta p(t)\|} \quad (11)$$

- iterate the process with $t = t+1$, while $\|\xi(t) - \xi^*\| > \epsilon$.

Similar rules were used when reaching was performed in the real end-effector configuration space \mathcal{C} using the analytic Jacobian J_G . We call the reaching trajectories 'F-based' when they are generated using the NN-estimated function F , and 'G-based' if the exact mapping G was used instead.

III. RESULTS

The learning and reaching algorithms described in II-A and II-C are applied to a simulated robotic arm in 2D space (see Fig. 2a). The robot is redundant and has access to three proprioceptive variables p_k when the end-effector has only two degrees of freedom (dof). Two scenarios are considered depending on the kind of exteroceptive information the robot has access to (described hereafter). For each scenario, the algorithms parameters are set to:

- threshold of the neighborhood function: $\mu = 0.1$.
- centering factor: $\gamma = 10^{-3}$.
- maximum number of learning iterations: $T = 1500$.
- minimal error value: $Q_{min} = 10^{-10}$.
- number of exploratory movements: $N = 1000$.
- reference proprioceptive configuration: $[m_1, m_2, m_3] = [\frac{\pi}{5}, -\frac{3\pi}{5}, \frac{3\pi}{5}]$ (rad).
- proprioceptive exploratory amplitude: $A = \frac{\pi}{2}$ (rad).
- proprioceptive amplitude for the Jacobian estimation and update step size: $\epsilon = 10^{-3}$.

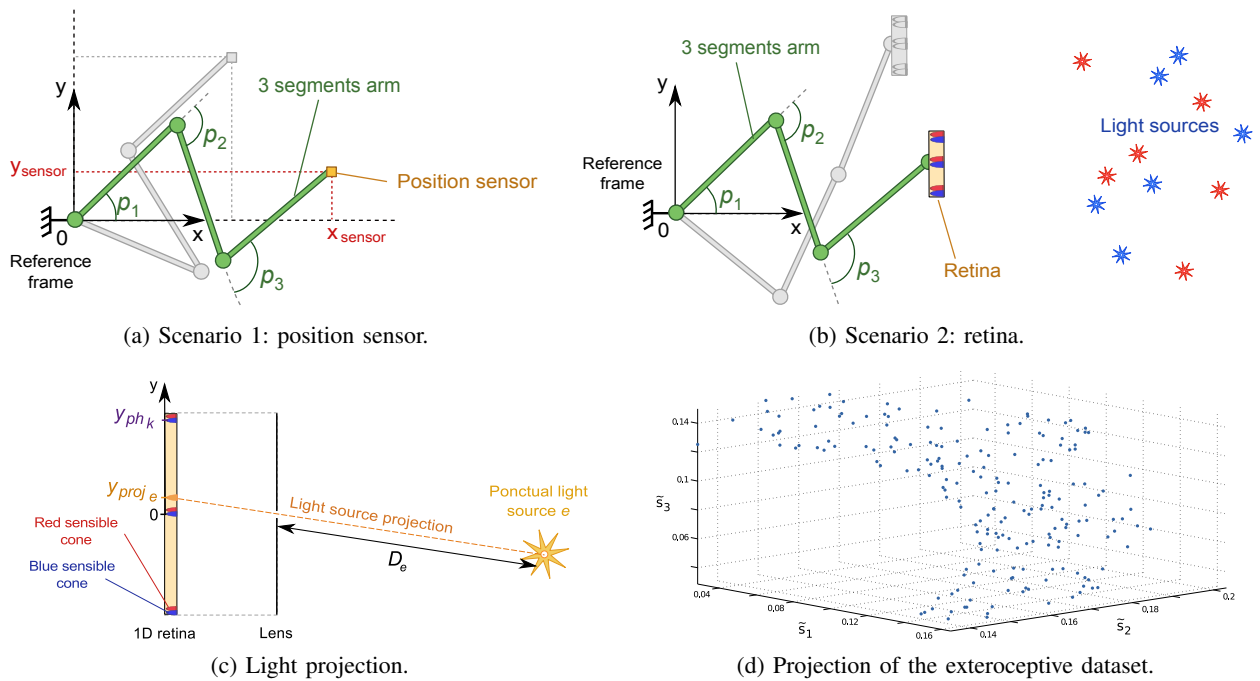


Fig. 2: The robotic arm is made up of three segments of equal length. The relative segment orientations are controlled by individual motors $\{m_1, m_2, m_3\}$. (a) Scenario 1: the end-effector is equipped with a position sensor. (b) Scenario 2: the end-effector is equipped with a retina-like sensor sensitive to colored light sources placed in the environment. (c) Each source is projected on the retina through a pinhole lens. The closer the source is to the retina and its projection is to a cone, the stronger the cone excitation. (d) Example of a 6D exteroceptive dataset projected on the 3 first coordinates of the exteroceptive space.

A. Scenario 1

In the first scenario, the robot is equipped with a position sensor providing the cartesian coordinates of its end-effector in absolute space. The one redundant dof of the arm implies that the null-spaces of J_F and J_G are lines in proprioceptive space \mathcal{P} from which angular divergence can be computed. On a set of 49 proprioceptive configurations p^i associated with regularly distributed end-effector positions c^i in the working space (see Fig. 3a and 3d), those null-spaces were close with an angular divergence of $1.5 \text{ deg} \pm 1.5 \text{ deg}$ (average \pm standard deviation).

Figure 3a presents several F -based and G -based reaching trajectories computed using the mappings F and G respectively. The target points for F - and G -based trajectories are defined in different spaces (Ξ and \mathcal{C} , respectively). To assure that they are consistent with each other, we first choose the target point c^* for the G -based trajectory, take a random robot configuration p^* corresponding to this point (i.e. $G(p) = c$), and then define the target ξ^* as $\xi^* = F(p^*)$ for F -based trajectory.

Note that the configuration p^* usually does not coincide with the final configuration of the arm for the F -based trajectory. Hence, if the mapping F were imprecise, the final points of the F - and G -based trajectory could be different. As Fig. 3a clearly shows, it is not the case here.

There is still a small divergence between the F - and G -based trajectories. The reason of it is that the spaces Ξ and

\mathcal{C} do not have the same metrics and hence a straight line in \mathcal{C} may correspond to a curve in Ξ and *vice versa*. Figure 3b shows the same reaching trajectories, but in the space of internal representation Ξ . It can clearly be seen that the F -based trajectories, which were curved in the space \mathcal{C} (Fig 3a), are straight in the space of Ξ , and that the opposite holds for G -based trajectories.

The proprioceptive (angular) profiles differed very slightly for F - and G -based trajectories. Figure 3c presents the worst case (corner-to-corner reaching) and still the blue and orange curves are barely distinguishable. Taken together, all presented evidence suggests that in scenario 1 the robot was able to learn the internal representation of its end-effector configuration space.

B. Scenario 2

In the second scenario, the robot is equipped with an array of photoreceptors resembling an animal's retina. To keep the dimensionality of the problem the same as in the first scenario we assume that the retina preserves its orientation in the absolute reference frame. Note that this assumption is made exclusively for the sake of comparison; a qualitatively similar behavior can be expected when the retina is allowed to rotate.

The environment is represented by numerous red and blue light sources placed in front of the robot (see Fig. 2b). The light sources are projected on the retina through a pinhole lens as illustrated in Fig. 2c. There are two kinds

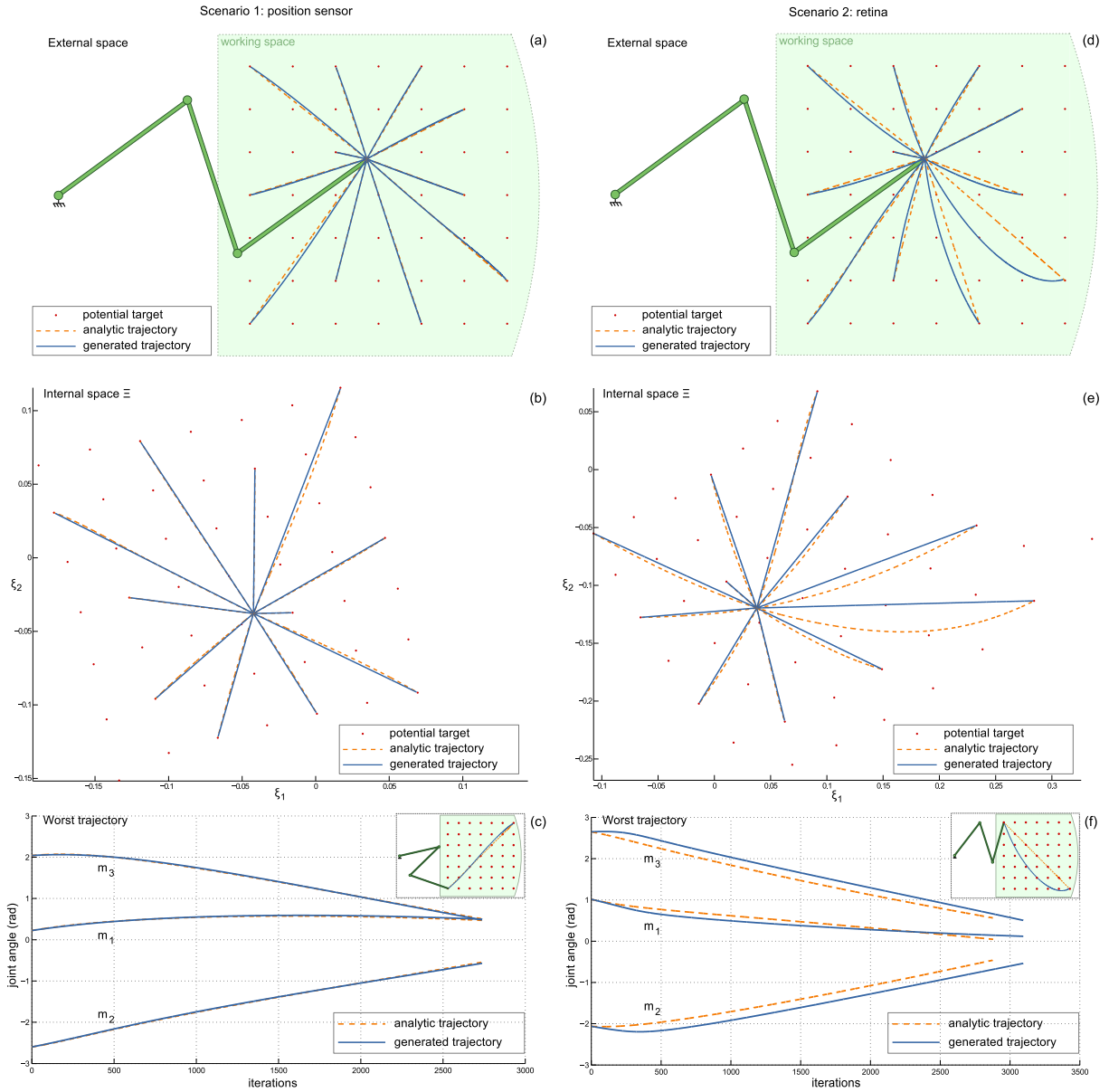


Fig. 3: (a)(d): G -based and F -based trajectories for several target configurations c^* distributed in the working space for scenarios 1 and 2 respectively. (b)(e): internal target states ξ^* associated with external configurations c^* and G -based and F -based trajectories in the internal space Ξ . (c)(f): proprioceptive profiles of the worst reaching case task displaying the greatest deviation of F -based trajectory from the G -based one.

of photoreceptors: sensitive to blue light and sensitive to red light. The retina has six photoreceptors, three of each kind. The response of the k -th photoreceptor to a single source e has Gaussian profile:

$$s_{k,e} = \frac{\exp(-\|y_{proj_e} - y_{ph_k}\|^2)}{D_e} \quad (12)$$

where y_{proj_e} is the position of the source e projection on the retina, y_{ph_k} the position of the photoreceptor k on the retina and D_e the distance between the lens' center and the source e . The excitation s_k of the k -th photoreceptor equals the sum of $s_{k,e}$ for all sources e of the corresponding color.

In this case the exteroceptor space is built on six photore-

ceptors and thus has dimension six. For illustration, a three-dimensional projection of the exteroception component of the training data is shown in Fig. 2d. In scenario 2, this cloud of points represents the information that the agent has about its end-effector position. One can notice that the points tend to form a surface. This happens because although the dimension of the exteroception space is six, there is only a two-dimensional manifold of different positions of the end-effector (retina). Clearly, from this cloud of points, the position of the end-effector cannot be easily extracted. Moreover, as the environment changes, so does the surface to which the points belong.

Figure 3d presents the F - and G -based trajectories gen-

erated for the same reaching targets, defined the same way as in scenario 1. Again, in spite of the fact that the angular configurations are different at the final points of both types of trajectories (see Fig. 3f), the final end-effector positions in the real space are the same. This shows that the mapping F correctly associates the configurations with the same retina position. Indeed, the null-space of the Jacobian J_F was close to the null-space of J_G : the angular difference between them was $4.8 \text{ deg} \pm 5.7 \text{ deg}$.

The F - and G -based trajectories have significantly different shapes. This result can be understood by looking at the projection of the potential target grid in the internal representation space Ξ (see Fig. 3e). The use of the retina implies that the metrics of Ξ , locally derived from exteroceptive metrics, is significantly different from the metrics in \mathcal{C} . The grid of targets ξ^* for F -based trajectory is non-uniform in the Ξ space.

The evolution of each proprioceptive component (angles) for the worst reaching case is presented in Fig. 3f. Here again, these proprioceptive changes correspond to the shortest angular trajectory for which the robot moves along a straight line in the representation space Ξ .

IV. CONCLUSION

In the current study we presented an algorithm that allows a robot to build an internal representation of its end-effector configuration space on the basis of exteroceptive data. The algorithm does not make any specific assumptions regarding the nature of the exteroception, nor does it require any prior knowledge about the robot's geometrical properties. The internal representation captures the topology of the real end-effector configuration space and preserves the invariance of the null-space of the Jacobian between proprioception and real end-effector position.

The internal representation can be used to control the robot and perform reaching tasks. Similar behavior could be achieved by learning directly the relation between the proprioceptive space and the exteroceptive space [1]. However this relation would vary for different environments. The mapping F could also be learned by projecting exteroceptive data in a low-dimensional representational space [4] and then estimating a mapping from the proprioceptive space to this projection. Both the projection and the resulting mapping would however depend on the environment. The NN we proposed overcomes these limitations. On the one hand, both the exteroceptive data projection and learning of the mapping are performed simultaneously through minimization of the cost function. On the other hand, the topology conservation criteria we introduced ensures that the internal representation is independent of the environment but captures the invariant component of successive explorations: the end-effector configuration.

In further development of this algorithm we plan to pay attention to the update of the pre-learned mapping following the change of the environment. If the statistics of the environment content is sufficiently large, this may help to reduce the metric distortion of the internal representation.

The algorithm is to be tested on a real robot. We expect difficulties to arise from elements of the robot structure interfering with exteroception. As a consequence, certain configurations can wrongly be classified as different.

REFERENCES

- [1] François Chaumette and Seth Hutchinson. *Visual Servoing*. Springer, 2008.
- [2] Alain Droniou, Serena Ivaldi, Vincent Padois, and Olivier Sigaud. Autonomous online learning of velocity kinematics on the iCub: A comparative study. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3577–3582. IEEE, 2012.
- [3] Alban Laflaquiere, Sylvain Argentiari, Olivia Breyse, Stéphane Genet, and Bruno Gas. A non-linear approach to space dimension perception by a naive agent. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3253–3259. IEEE, 2012.
- [4] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer, 2007.
- [5] Simon McGregor, Daniel Polani, and Kerstin Dautenhahn. Generation of tactile maps for artificial skin. *PLoS one*, 6(11):e26561, 2011.
- [6] D Philipona, J K O'Regan, and J.-P. Nadal. Is there something out there?: Inferring space from sensorimotor dependencies. *Neural Comput.*, 15(9):2029–2049, 2003.
- [7] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591. IEEE, 1993.
- [8] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [9] Thomas Schatz and P-Y Oudeyer. Learning motor dependent Crutchfield's information distance to anticipate changes in the topology of sensory body maps. In *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*, pages 1–6. IEEE, 2009.
- [10] Olivier Sigaud, Camille Salaün, and Vincent Padois. On-line regression algorithms for learning mechanical models of robots: a survey. *Robotics and Autonomous Systems*, 59(12):1115–1129, 2011.
- [11] Joshua Tenenbaum, Vin de Silva, and John Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, December 2000.