

ONLINE LEARNING OF THE BODY SCHEMA

MICHA HERSCH*, ERIC SAUSER† and AUDE BILLARD‡

*LASA, School of Engineering, EPFL,
Station 9, 1015 Lausanne, Switzerland*
**micha.hersch@epfl.ch*
†*eric.sauser@epfl.ch*
‡*aude.billard@epfl.ch*

Received 13 July 2007
Accepted 28 January 2008

We present an algorithm enabling a humanoid robot to visually learn its body schema, knowing only the number of degrees of freedom in each limb. By “body schema” we mean the joint positions and orientations and thus the kinematic function. The learning is performed by visually observing its end-effectors when moving them. With simulations involving a body schema of more than 20 degrees of freedom, results show that the system is scalable to a high number of degrees of freedom. Real robot experiments confirm the practicality of our approach. Our results illustrate how subjective space representation can develop as a result of sensorimotor contingencies.

Keywords: Kinematic learning; tool use adaptation; body schema; peripersonal space representation; multimodal integration; developmental robotics.

1. Introduction

One of the major components of human agility and dexterity is arguably its ability to merge multiple sensory informations and learn the relationships between informations coming from different modalities. In traditional robotics, the importance of learning the mapping between different sensory modalities is often neglected. The robot geometry is generally assumed to be known, and the sensor properties are acquired offline, via a calibration process.

For humanoid robots, which usually comprise a high number of degrees of freedom (DOFs) and of sensors, and are intended to operate in human environments, learning the sensorimotor contingencies is likely to bring significant advantages, as it amounts to a continuous self-calibration. For example, by learning its body schema, a humanoid robot can adapt to errors in the vision calibration, or to the use of tools. It makes the robotic system less dependent on the human measurements of the system, such as body shape, visual marker positions, and so on.

On the other hand, it has been argued that humanoid robots offer an interesting platform for testing hypotheses on human cognition.²⁸ Despite their limitations, humanoid robots can be taken as (rough) models of human bodies in that they

have multiple DOFs and multiple sensory inputs. It is then possible to implement computational processes putatively taking place in the brain and see how they perform for humanoids. This is likely to yield interesting indications of the effectiveness of those hypothesized computational processes.

Bearing in mind this twofold approach, we propose a mechanism by which a humanoid robot can learn its body schema through combining information from the proprioception (motor encoders), the stereovision and possibly tactile sensors. The body schema is modeled by a hierarchy of frames of reference (FOR) transformations which are continuously adapted as sensory information is acquired by the robot.

The rest of this paper is organized as follows. Section 2 provides a succinct overview of previous approaches to learning the body schema. Section 3 recalls some well-known facts about how multisensory information is integrated in primates to form a coherent view of the peripersonal space. Section 4 introduces a new model of an adaptive body schema and provides an algorithm for learning it. In Sec. 5, experiments are performed first in simulation on a 24-DOF humanoid robot and then with a real humanoid robot that adapts to the use of a tool. Those results are discussed in Sec. 6, which also highlights the relevance of the model to the study of human cognition. A brief conclusion is presented in Sec. 7.

2. Related Work

The robot geometry is the key element which determines the forward and inverse kinematic functions of this robot. The forward kinematic function K is defined by the relationship between the vector of joint angles \mathbf{q} defining a given arm configuration and the corresponding position \mathbf{x} of the end-effector in space:

$$\mathbf{x} = \mathbf{K}(\mathbf{q}). \quad (1)$$

The inverse kinematic function \mathbf{K}^{-1} is the inverse function, $\mathbf{q} = \mathbf{K}^{-1}(\mathbf{x})$. Finding this inverse function is an ill-posed problem in the case of redundant manipulators. Most work dealing with the learning of a robot geometry directly tackles the issue of learning the inverse kinematic function. This is indeed one of the standard issues in robotics — namely, knowing that we want the robot to reach a particular location \mathbf{x} in space, what are the arm configurations \mathbf{q} (the vector of arm joint angles) that bring the end-effector to this location? This is a fundamental issue, and many solutions have been suggested. If the forward kinematic function \mathbf{K} is known, it is possible to use local solutions that iteratively bring the robot end-effector to the desired location by computing some (pseudo)inverse of the Jacobian matrix of \mathbf{K} .^{23,29,32,34}

If \mathbf{K} is unknown, there are global solutions that directly learn the mapping \mathbf{K}^{-1} between the end-effector position and the corresponding joint angles. This is typically done using some function approximators, like multilayer perceptron,^{12,20} locally weighted projection regression,⁷ self-organizing maps,^{10,31} possibly combined with quantum clustering.¹⁷

Finding the forward kinematic function is much easier. It can be done by measuring the segment lengths and computing the chain of successive rotations and translations (as done in Subsec. 4.1). Therefore, it has not attracted much attention in the robotics community. There are, however, some works, mostly coming from the epigenetic robotics community, where an artificial system learns to control its motions. For example Kuperstein¹⁸ learns a visuomotor coordination, and in Metta *et al.*²² the motor torques for reaching a point are learned. In those examples, the number of DOFs of the arm is generally quite low (between two and five). Using self-organizing maps, Fuke *et al.*⁹ learn the correspondence between visual, proprioceptive and tactile informations in a simulated arm–face system.

The work presented here differs from previous approaches in that the learning is performed entirely online and it can deal with a high number of DOFs. Moreover, the model does not focus solely on determining the position of the end-effector, but also yields the position of each segment and can compute the associated Jacobians. It thus provides additional information, which can be very useful, such as for obstacle avoidance or for computing iterative local inverse kinematics. Finally, the system presented here can offer interesting explanations of the way humans represent their peripersonal space, in a multimodal way.

3. The Body Schema

In this section we review some evidence based on psychophysical and neurophysiological studies that suggest that, in primates, multisensory information is integrated through a hierarchy of FORs that reflects the body structure. This hierarchy allows a mapping across the visual, proprioceptive, motor and tactile modalities and is highly adaptive. A more comprehensive review has been written by Holmes and Spence.¹⁴

There is undoubtedly a strong interaction between visual, tactile and proprioceptive sensory informations. The existence of bimodal visuotactile neurons in the monkey and psychophysical experiments involving cross-modal extinctions have put in evidence the existence a visuotactile representation,¹⁹ while the discovery of body-part centered visual fields²⁷ shows that there is a strong interaction between proprioception and vision.

It is believed that each sensory–motor modality receives and provides information represented within different FORs. For instance, proprioception, touch and motor commands are coded in a FOR centered on the specific body part they represent and control,^{1,26,30} i.e. a local representation, whereas visual information is perceived in an eye-centered or retinotopic manner,¹⁶ i.e. a visual representation. Since visual, tactile and proprioceptive feedback are tightly coupled in space and time and together form the representation of one’s own body, multisensory information has to be integrated across modalities so that a coherent view of the body can emerge. It has been suggested that this integration is made through a series of

transformations across intermediary FORs.^{6,25} Indeed, neurons coding position in a FOR centered on body parts have been reported by Graziano and Gross.¹¹

The adaptivity of those transformations is particularly evident in psychophysical experiments involving prism adaptation.³³ It has long been known that when subjected to a visual shift or distortion caused by a prism, human subjects first tend to reach, expectedly, the seen position, rather than the actual position of the reaching target. After a while, however, they can correct for the visual distortion and accurately reach the target. For this to occur, visual and proprioceptive feedback of the hand is necessary. When the visual distortion is removed, the subjects show so-called after-effects, i.e. they still reach the virtual target, as if the visual distortion was still active. This occurs although they are aware that it is not the case. This adaptability has been demonstrated for visual shifts (rotations), reflections and stretches but could not be shown for more complicated deformations which do not preserve the space topology.³ Furthermore, people could adapt to transformations expressed in intrinsic (joint angle) coordinates.¹⁵

Another kind of experiment emphasizing the adaptiveness of the body schema involves the use of tools. It has been shown that after some practice with a tool, the monkey integrates this tool into his body schema.²¹ The somatosensory receptive field of given neurons was observed to be expanded by the tool, after some practice.

Finally, the “fake limb” experiments also highlight the adaptive and tight connection between different sensory modalities and the feeling of one’s own body.⁴ In those experiments, a subject sees a fake limb being touched synchronously with his real, unseen arm and feels that the fake arm is his.

This argues in favor of the existence of a comprehensive framework, which allows one to combine information across visual, tactile and proprioceptuomotor modalities, and to perform the appropriate FOR transformations required for the integration of this information. Those transformations are highly adaptive, and are constantly learned as a result of sensory experience.

4. A Model of the Adaptive Body Schema

4.1. *Kinematic chains*

Considering a serial manipulator with n rotative joints, it is possible to compute how a position given in the end-effector FOR can be expressed in the manipulator base FOR. In other words, we can compute the FOR transformation from a FOR centered on the distal segment to the FOR centered on the proximal segment. This is done by considering the rotation and translation corresponding to each joint and segment, as is commonly done for computing kinematic functions, e.g. the Denavit–Hartenberg kinematic chain parametrization. This transformation can be seen as a series of successive rotations and translations, where the rotation angles are given by the manipulator joint angle and the translations are given by the vector difference between the joints. Thus, it is possible to transform a vector \mathbf{v}_n from a FOR centered on the end-effector to a vector \mathbf{v}_0 in a FOR centered on the other side of the chain

by a transformation \mathcal{T} described by the following equation:

$$\begin{aligned} \mathbf{v}_0 &= \mathcal{T}(\mathbf{v}_n) \\ &= \mathbf{T}_1 \circ \mathbf{R}_1 \circ \mathbf{T}_2 \circ \mathbf{R}_2 \circ \dots \circ \mathbf{T}_n \circ \mathbf{R}_n(\mathbf{v}_n) \\ &= \mathbf{l}_1 + \mathbf{R}_1(\mathbf{l}_2 + \mathbf{R}_2(\dots(\mathbf{l}_n + \mathbf{R}_n(\mathbf{v}_n))\dots)), \end{aligned} \tag{2}$$

where \mathbf{T}_i and \mathbf{R}_i represent respectively the translation and rotation corresponding to segment i and joint i , \mathbf{l}_i denotes the vector representing the link proximal to joint i at the zero position, and \mathbf{R}_i is the rotation caused by joint i . Figure 1 illustrates how the segments are numbered and how this FOR is computed. Note that, similarly to the Denavit–Hartenberg parametrization of kinematic chains, \mathbf{l}_i can be zero if joints $i - 1$ and i have the same rotation center.

4.2. Single segment adaptation

We consider the following problem regarding a single joint manipulator (see Fig. 2). We assume that we have an initial guess of the unit rotation axis \mathbf{a} and the joint position \mathbf{l} . Now, given a vector \mathbf{v} in a FOR centered on the distal segment, its actual transform \mathbf{v}' centered on the proximal segment and the rotation angle θ , how is it possible to adapt \mathbf{a} and \mathbf{l} so that they account better for the actual transformation induced by the manipulator?

In order to do so, we perform a simple gradient descent on the squared distance between the actual and the simulated transform vector:

$$\Delta \mathbf{l} = -\epsilon \frac{\partial}{\partial \mathbf{l}} \frac{1}{2} \|\mathbf{v}' - (\mathbf{l} + \mathbf{R}_\mathbf{a}^\theta(\mathbf{v}))\|^2, \tag{3}$$

$$\Delta \mathbf{a} = -\epsilon \frac{\partial}{\partial \mathbf{a}} \frac{1}{2} \|\mathbf{v}' - (\mathbf{l} + \mathbf{R}_\mathbf{a}^\theta(\mathbf{v}))\|^2, \tag{4}$$

where $\mathbf{R}_\mathbf{a}^\theta$ is the rotation of the angle θ around the axis \mathbf{a} and the learning step ϵ is a small positive scalar. The derivative with respect to \mathbf{l} in Eq. (3) is straightforward

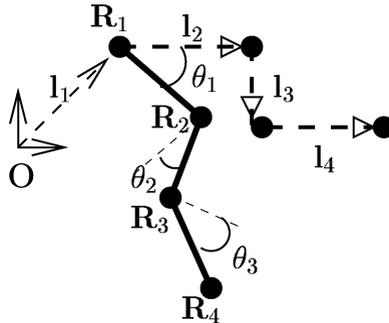


Fig. 1. The parametrization of a kinematic chain. The dashed line represents the kinematic chain in the zero position (when all angles are equal to zero). The solid line represent the same chain with different rotation angles. O refers to the origin.

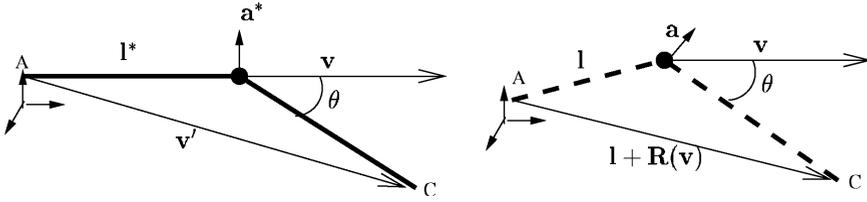


Fig. 2. The learning problem for a single segment. The real rigid transformation is shown on the left and is parametrized by unknown vectors \mathbf{l}^* and \mathbf{a}^* and known angle θ . The current guess of this rigid transformation appears on the right and is parametrized by \mathbf{l} , \mathbf{a} and θ . Knowing a vector \mathbf{v} and its real transform \mathbf{v}' and the rotation angle θ , we try to update our guess of \mathbf{l} and \mathbf{a} . The letters A and C indicate respectively the origin and the end-effector of the manipulator.

to compute. We have

$$\Delta \mathbf{l} = \epsilon(\mathbf{v}' - (\mathbf{I} + \mathbf{R}_{\mathbf{a}}^{\theta}(\mathbf{v}))). \quad (5)$$

In order to compute the derivative with respect to \mathbf{a} in Eq. (4), we make use of the Rodrigues formula²:

$$\mathbf{R}_{\mathbf{a}}^{\theta}(\mathbf{v}) = \cos(\theta)\mathbf{v} + \sin(\theta)\mathbf{a} \times \mathbf{v} + (1 - \cos(\theta))\mathbf{a}^T \mathbf{v} \mathbf{a}. \quad (6)$$

Hence

$$\bar{\mathbf{R}}_{\mathbf{a}}^{\theta} \doteq \frac{\partial}{\partial \mathbf{a}} \mathbf{R}_{\mathbf{a}}^{\theta}(\mathbf{v}) = \sin(\theta)\mathbf{v} \uparrow + (1 - \cos(\theta))(\mathbf{a}\mathbf{v}^T + (\mathbf{a}^T \mathbf{v})\mathbf{I}), \quad (7)$$

where \mathbf{I} is the 3×3 identity matrix and the unitary operator \uparrow is defined as

$$\mathbf{v} \uparrow \doteq \frac{\partial}{\partial \mathbf{a}} (\mathbf{a} \times \mathbf{v}) = \begin{pmatrix} 0 & v_3 & -v_2 \\ -v_3 & 0 & v_1 \\ v_2 & -v_1 & 0 \end{pmatrix}, \quad \text{with } \mathbf{v} = [v_1 \ v_2 \ v_3]^T. \quad (8)$$

Thus

$$\Delta \mathbf{a} = \epsilon(\mathbf{v}' - (\mathbf{I} + \mathbf{R}_{\mathbf{a}}^{\theta}(\mathbf{v})))^T (\sin(\theta)\mathbf{v} \uparrow + (1 - \cos(\theta))(\mathbf{a}\mathbf{v}^T + (\mathbf{a}^T \mathbf{v})\mathbf{I})). \quad (9)$$

Since \mathbf{a} must be of unit norm, it is normalized to 1 after being updated. This solves our problem. Using Eqs. (5) and (9), it is possible to adapt the representation of the joint position and orientation online, as examples of positions in the distal FOR and the corresponding position in the proximal FOR are provided. This algorithm always converges to the correct translation and rotation axis when provided with enough different values of \mathbf{v} and \mathbf{v}' (see Subsec. 4.4).

4.3. Multisegment adaptation

We can now apply the same principle to multisegment manipulators. Starting from Eq. (2), it is possible to compute

$$\Delta \mathbf{l}_i = -\epsilon \frac{\partial}{\partial \mathbf{l}_i} \frac{1}{2} \|\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n)\|^2, \quad (10)$$

$$\Delta \mathbf{a}_i = -\epsilon \frac{\partial}{\partial \mathbf{a}_i} \frac{1}{2} \|\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n)\|^2, \quad (11)$$

where \mathbf{a}_i is the rotation axis of \mathbf{R}_i . If \mathbf{R}_i is the rotation matrix corresponding to joint i (i.e. of axis \mathbf{a}_i and angle θ_i), we have

$$\frac{\partial}{\partial \mathbf{l}_i} \mathcal{T}(\mathbf{v}_n) = \prod_{j=1}^{i-1} \mathbf{R}_j, \quad (12)$$

$$\frac{\partial}{\partial \mathbf{a}_i} \mathcal{T}(\mathbf{v}_n) = \left(\prod_{j=1}^{i-1} \mathbf{R}_j \right) \frac{\partial}{\partial \mathbf{a}_i} (\mathbf{R}_i(\mathbf{T}_{i+1} \circ \mathbf{R}_{i+1} \cdots \circ \mathbf{T}_n \circ \mathbf{R}_n(\mathbf{v}_n))), \quad (13)$$

where the derivative on the right-hand side of the latter equation is obtained by applying Eq. (7). All the rotation axes and translation vectors can thus be simultaneously updated using

$$\Delta \mathbf{l}_i = \epsilon (\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n))^T \prod_{j=1}^{i-1} \mathbf{R}_j, \quad (14)$$

$$\Delta \mathbf{a}_i = \epsilon (\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n))^T \left(\left(\prod_{j=1}^{i-1} \mathbf{R}_j \right) \frac{\partial}{\partial \mathbf{a}_i} (\mathbf{R}_i(\mathbf{T}_{i+1} \circ \mathbf{R}_{i+1} \cdots \circ \mathbf{T}_n \circ \mathbf{R}_n(\mathbf{v}_n))) \right). \quad (15)$$

4.4. Convergence

4.4.1. Single joint case

Theorem 1. *Assuming that we run the algorithm on a set of configurations given by $\{\mathbf{v}, \mathcal{T}^*(\mathbf{v}), \theta_j\}_{j=1}^J$, where θ_j follows a symmetric probability density function (pdf) centered on 0, such that $\text{var}(\cos \theta_j) \leq 2\text{var}(\sin \theta_j)$, the algorithm described by iteratively applying (3) and (4) converges to a correct estimate of \mathbf{a} and \mathbf{l} .*

The proof is given in the appendix.

4.4.2. Multisegment case

The convergence for the multisegment case cannot be proven. In order to have an idea of the convergence properties, simulations were performed. In a single simulation run, the rotation axes \mathbf{a}_i^* and \mathbf{a}_i of two kinematic chains were randomly generated. The \mathbf{l}_i were initialized with small random values and the algorithm was run in order to see whether the \mathbf{a}_i converge to the \mathbf{a}_i^* . Convergence is considered to be attained if the distance between the real limb position and the modeled limb position remains smaller than a threshold (around 1% of chain length) over 500 different configurations. Thousands of those runs were performed for 1-, 3-, 5- and 7-DOFs kinematic chains. The results can be seen in Fig. 3 (left). Expectedly, if there is only one joint, the algorithm always converges. When there are more DOFs, the algorithm fails to converge after a million iterations in less than 1% of the cases. The time it takes for convergence is plotted in Fig. 3 (right). Figure 4 gives an example of the evolution of the estimate of the rotation axes for a kinematic chain containing 3 DOFs.

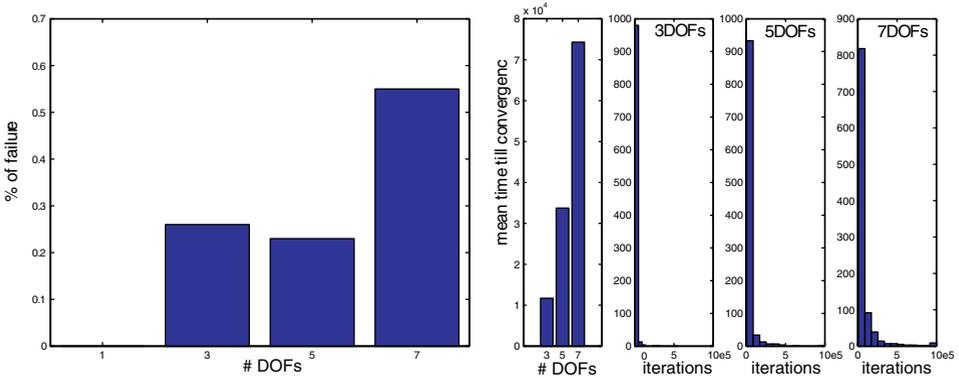


Fig. 3. Left: The percentage of trials that did not converge after a million iterations. Right: The time needed for convergence depending on the number of DOFs. The bars on the left show the mean number of iterations until convergence, and the three histograms on the right show the distributions of convergence time. The distributions have quite a long tail, indicating that in some cases it takes much longer than average to converge.

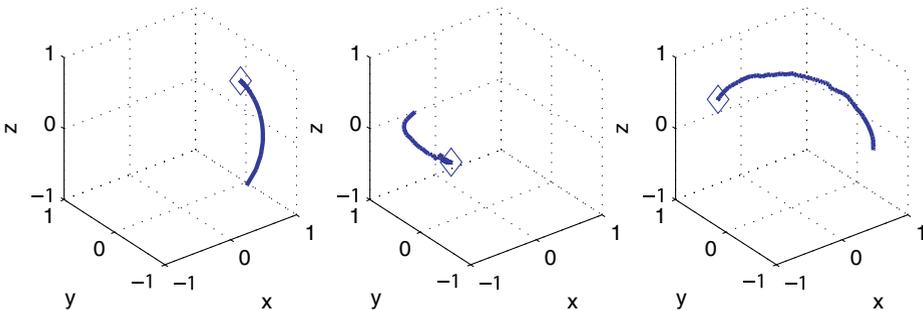


Fig. 4. The evolution of the rotation axes for a 3 DOF kinematic chain. The real axes are indicated by the three diamonds, and each graph shows the evolution of one estimated axis on a sphere of radius 1.

4.5. Adaptive body schema

4.5.1. Kinematic tree

The humanoid body schema can be represented as a tree of rigid transformations reflecting the limb structure, as shown in Fig. 5. We thus have a kinematic tree with adaptive joint positions and orientations. Note that the structure of this tree (i.e. the number of joints and the ordering) is given and remains fixed.

Out of this tree, kinematic chains can easily be extracted as paths in the tree. It is possible to compute the FOR transformation from a FOR attached to any joint of the kinematic tree to a FOR attached to any other joint. This is done by first finding the path joining the two corresponding nodes. To each edge along this path, there corresponds a FOR transformation. Depending on the direction in which an edge

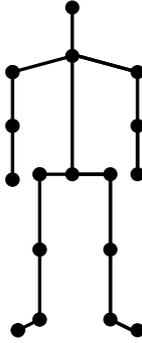


Fig. 5. A kinematic tree representing a humanoid. Nodes represent rotations and edges represent translations.

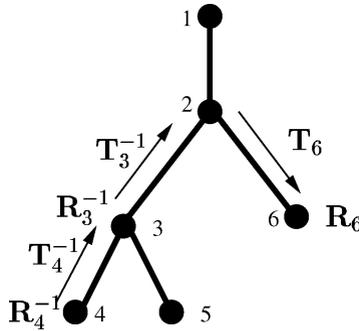


Fig. 6. The generating of the kinematic chain for the FOR transformation relating the FOR attached to two nodes in the tree. When taking an edge up (to the root), one takes the inverse transform, and when taking an edge down (to the leaf), the actual transform is taken. In this example, the transformation from a FOR centered on joint 6 to a FOR centered on joint 4 is given by $\mathbf{R}_4^{-1} \circ \mathbf{T}_4^{-1} \circ \mathbf{R}_3^{-1} \circ \mathbf{T}_3^{-1} \circ \mathbf{T}_6 \circ \mathbf{R}_6$.

is taken, the transformation or its inverse is considered. An example is provided in Fig. 6, showing how a kinematic chain is extracted from the kinematic tree.

4.5.2. Body schema learning

We assume that the robot is endowed with a stereovision system that can track the position of its end-effectors. This position is provided in a head-centered FOR. Within the kinematic tree, the path going from the head to the end-effector corresponds to a kinematic chain that transforms positions and orientations from a FOR centered on the end-effector to a visual or head-centered FOR. Using (15) and (14), it is possible to update all the rigid transformations along this chain. As input \mathbf{v}'_n are given by the stereovision system and \mathbf{v}_n is the position of the end-effector in its own FOR. This is illustrated in Fig. 7 (left). This figure also shows how the same algorithm could be used with tactile sensors for tactile body schema learning.

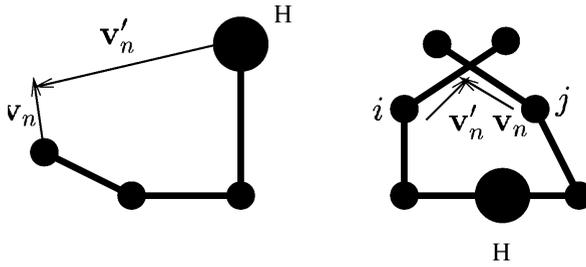


Fig. 7. Humanoid body schema adaptation. The big circle marked H is the head of the humanoid. Left: Visual learning side view of the robot. The position of the limb in its own FOR \mathbf{v}_n is transformed into a visual head-centered FOR \mathbf{v}'_n . Right: Tactile learning, top-view. The positions of touch sensors in the FOR of their limbs \mathbf{v}_n and \mathbf{v}'_n transform into one another.

4.5.3. Subjective body schema

Traditionally, the body schema has been considered as an objective account of the body characteristics, such as the arrangement of the limbs, their lengths or the positioning and effect of the joints. It is possible to oppose this “objective” body schema with a “subjective” view of the body schema, which would be dependent on the perceptual abilities of the robot. In this view, which is adopted in this paper, the body schema deals only indirectly with physical properties of the body. It primarily deals with the FOR transformations associated with the sensory signals. For example, given a proprioceptive input corresponding to a particular posture, the body schema can predict the corresponding visual perception. This depends not only on the physical properties of the body but also on the properties of the sensory system. Moreover, it can be that a precise account of the physical properties of the body is not necessary for the “subjective body schema,” depending on the sensory system. For example, in our case, if the robot can track only end-effector positions, many different body geometries will yield the same “subjective” body schema. In the simple example depicted in Fig. 8, such a robot will not be able to differentiate between the two “objective” body schemas. These will correspond to the same “subjective” body schema. As the corresponding kinematic function and

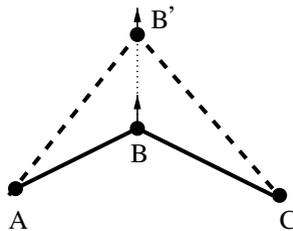


Fig. 8. A simple example of two different geometries (solid and dashed) yielding the same FOR transformation from an A-centered to a C-centered FOR.

Jacobian remain the same in both cases, using one or the other body schema for controlling its movements will produce the same end-effector trajectories.

5. Humanoid Experiments

5.1. Setup

In order to validate the algorithm described above, we tested it in simulation on a 24 DOF humanoid robot. Testing the algorithm on a real robot with that many DOFs would be quite impractical, for the following reasons.

- It requires the availability of such a sophisticated robot, with that many DOFs, stereovision and a tactile skin.
- The high number of positions that the robot has to visit would require from a real robot many days of continuous exploration for the algorithm to converge.
- A real robot may not be able to see all its end-effectors (in particular its feet), due to a limited visual angle and joint range and possible occlusions.

The simulated humanoid (or avatar) has the shape of the Fujitsu Hoap3 robot and comprises 24 DOFs. A schematic of the robot is drawn in Fig. 9. When learning the body schema, the avatar configuration space was randomly sampled with a uniform distribution. The joint angles and corresponding visual position were fed into the

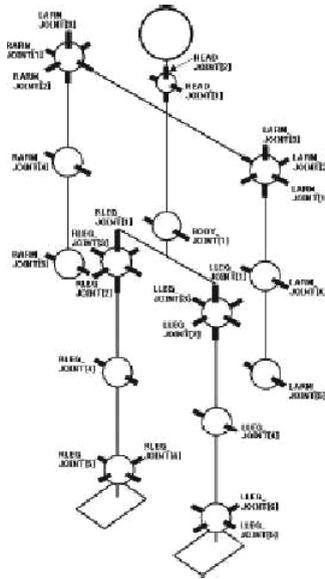


Fig. 9. The structure of the Hoap3 robot. Spheres show the joints, with the rotation axis shown as dark lines going through them. The hand rotations were not used, and an additional head roll joint was modeled. This makes 24 DOFs. This picture is from the Hoap documentation provided by Fujitsu.

algorithm. The body schema was initialized with random joint orientations and small random body segments.

5.2. Results

In this section, we present a set of experiments intended to first validate the framework described above. In a first validation step, we used the algorithm described above to learn the body schema of the Hoap3 robot described in Subsec. 5.1. In this first experiment, only the two hands and feet were tracked. This means that four kinematic chains were concurrently used: head–right hand, head–left hand, and head–right foot, and head–left foot. At each time step a joint angle configuration was randomly chosen, and the corresponding positions of the hands and feet in a head-centered FOR were computed for the Hoap3 robot. Along with the joint angle values, those four positions were fed as \mathbf{v}'_n [see Eqs. (14) and (15)] for adapting the corresponding kinematic chains. The result can be seen in Fig. 10, which plots the error of the kinematic function, i.e. $\|\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n)\|$, at each iteration. After many iterations, the system converges to the appropriate subjective body schema as the error converges to zero. Note, however, that the objective body schemas differ, as can be seen in Fig. 11.

In this experiment, a minimal amount of information is provided by the vision system as it always tracks only end-effectors, like hands and feet. But it is also possible for the vision system to track nonterminal body parts, like elbows and

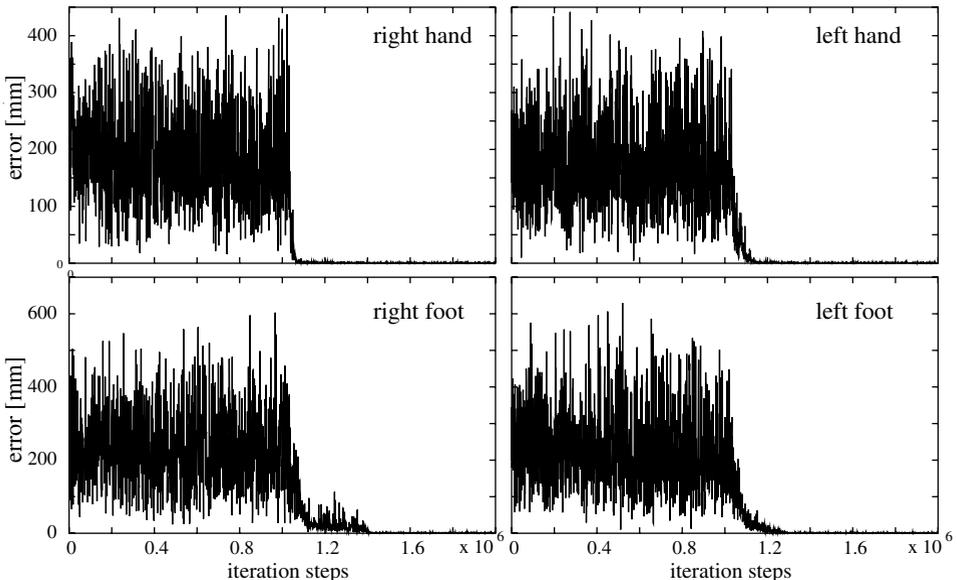


Fig. 10. The convergence of the learning algorithm. On the y -axis is the error in the computation of the limb position, in a head-centered FOR.



Fig. 11. Left: The Hoap3 “real” body schema, when standing on its knees. Middle: The objective body schema that was learned by the system when looking only at its hands and feet. It is not exactly the same, but the sensory relationships of both schemas (i.e. the subjective body schemas) are the same, when considering only vision of the hands and feet. Right: The body schema learned when also looking at elbows, knees, shoulders and the waist. The darker sticks indicate the rotation axis of each joint. There are three DOFs at the shoulders and the hips.

knees, as well. This, of course, is expected to make the system converge to the right geometry and also to considerably speedup the learning process, as much more information is available to the system. Indeed, tracking nonterminal body parts amounts to having shorter kinematic chains which significantly reduces the dimension of the problem.

So, in a second experiment, the vision system alternatively tracks terminal (hand and feet) and nonterminal (knees, elbows, shoulders and waist) body segments. In this case, convergence is faster and the robot geometry is correctly retrieved as there is no ambiguity on the joint locations (see Fig. 11).

5.3. Real robot experiment

In order to evaluate the practicality of our approach, we conducted an experiment in a real robotic setting. In this experiment, the robot carries an unknown tool. By looking at the tip of the tool the robot can integrate this tool into his body schema, thus enabling it to manipulate it adequately. This setting is shown in Fig. 12. We use the Hoap3 robot, which is endowed with a stereovision system.

In the results presented here, the robot is initialized with its “real” body schema. It holds a 340-mm-long stick in the hand, with a color blob at its tip. We then make the robot passively move its arm, with the tip of the stick remaining within the field of view of the cameras. The stereovision system tracks the tip of the stick, and the robot joint angles are read from the motor encoders. Those two sets of values (position of the tip of the stick and joint angles value) are then continuously fed into the learning algorithm, like in the simulation experiments. The vector \mathbf{v} in Eqs. (14) and (15) is the position of the end-effector in its own FOR; in other words, zero. Two cases (using the same data) are tested. In the first case, only the terminal limb is adaptive, i.e. we do not change the position and direction of the nonterminal joints. In the second case, the whole arm is adaptive, as in the

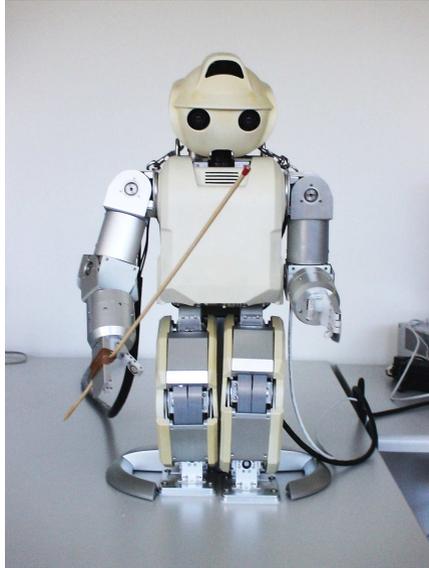


Fig. 12. The setup of the real robot experiment. The robot holds a stick and visually tracks its tip and records its arm joint angle values, while its arm is passively moved.

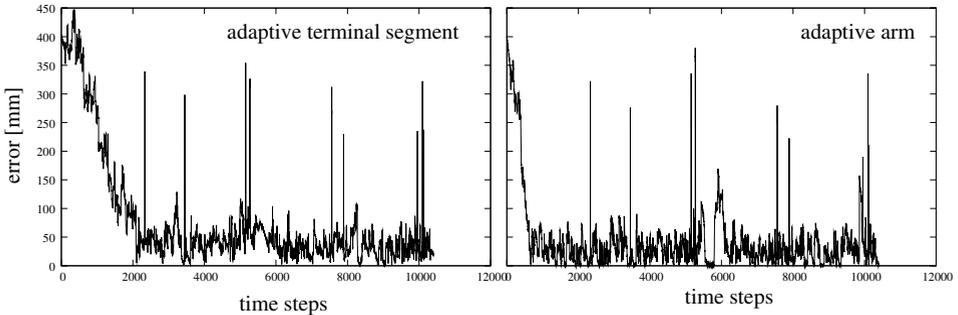


Fig. 13. Incorporating the tool in the body schema. These graphs show the evolution of the distance between the end-effector positions seen by the hand on one hand, and computed using proprioception and the body schema on the other hand. On the left, only the terminal limb is adaptive; on the right, all the limbs are adaptive. In both cases the body schema adaptation reduces the error from approximately 40 cm to 5 cm.

simulations presented above. The results are displayed in Fig. 13. This figure shows the distance between the positions of the end-effector as seen by the stereovision on one hand, and as computed by the body schema on the other hand. In other words, the y -axis plots $\|\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n)\|$, where \mathbf{v}'_n is given by the stereovision and \mathbf{v}_n is zero. One can see that in both cases the system starts with an error of about 40 cm, which corresponds approximately to the length of the stick, and reduces this error to about 5 cm. This means that the stick has been incorporated in the body schema, although

imprecision in the stereovision system keeps this error of about 5 cm. In the plot, one can notice a few peaks of large errors. These are outliers of the stereovision system, and correspond to a bad tracking of the color blob. Of course, the resulting error values are inevitable, but it is interesting to note that the system is quite robust to such outliers, as the next error values are again in a reasonable range. It takes about 2000 steps to reduce this error if only the terminal limb transformation is adaptive, and 1000 steps if all the limb transformations are adaptive. This takes 2–3 min, as updates are performed at a rate of approximately 10 H.

6. Discussion

Our results show that the suggested framework for body schema learning is effective. Indeed, simulation results show that it can learn the structure of a 24 DOF robot by tracking only the end-effectors. Furthermore, real robot experiments show that the method is applicable in a real setting. Given the fast development of actuator and sensor technologies, it may be possible in the near future to use this algorithm with all DOFs of real humanoid robots. For practical purposes, as it may be cumbersome to gather millions of points in the training set, it may be more efficient to apply this algorithm to a smaller set of data, well distributed in the joint ranges, and perform the iterations on this more restricted set.

The model presented here differs from earlier work on this topic mainly in that the knowledge of the kinematic structure is given in advance. In other words, the robot knows not only the number of DOFs, but also how they are arranged (serially or in parallel). Moreover, the model takes explicit advantage of the fact that those are rotative joints, which was usually not done in other works. Thus the effectiveness of the learning algorithm relies on this *a priori* knowledge. However, we believe that it is reasonable to assume such an *a priori* knowledge, as the kinematic structure of the humanoids usually does not change over time. Similarly, the kinematic structure of humans is fixed and does not evolve. Limbs grow, but new joints do not appear in a lifetime.

This last point illustrates that the results presented in this paper shed light on several interesting questions related to the learning of the body schema in robots and humans. In its present state our model cannot learn a new body structure, and to our knowledge it is unknown whether humans can adapt to different body structures. Another point raised by our results is that some body structures seem to be easier to learn than others, irrespective of the number of joints. It would be very informative to investigate whether this is an artifact of the learning procedure, or whether there is some intrinsic complexity dependent on the succession of rotation axes. Such various degrees of complexity have been described within the context of manipulator inverse kinematics,⁵ where manipulators with the same number of DOFs can have a different number of self-motion manifolds. If this complexity is intrinsic, it would be interesting to look at the complexity of the human body schema.

Because this model of the body schema consists in a hierarchy of coordinate system transformations, it can also be seen as a model of peripersonal space representation. Although the concepts of body schema and peripersonal space representation have traditionally been considered separately, the recent theories of motor perception⁸ and of a sensorimotor account of consciousness²⁴ suggest that those two concepts should be unified in a single framework. Indeed, perception and action are densely intertwined processes,¹³ which cannot be easily isolated one from another. However, within this work, we have not yet integrated the motor modality in our framework. This can be done by deriving the Jacobian from the body schema, by differentiating (2) with respect to the angles. This is one of the next steps intended for a further improvement of this model.

Another intended development is to extend the implementation of this model on the humanoid robot, and try to learn the whole body schema in a real setting. This is certainly not an easy task, as explained in Subsec. 5.1. It is interesting to note that most of the difficulties mentioned there are alleviated when considering babies learning their body schema. Indeed, they have stereovision and a tactile skin, it takes them several months, to learn their body schema, they are very flexible and thus have no problem in seeing their feet, hands and legs, and their compliant limbs allow them to explore their workspace much less cautiously (and thus faster) than the robot is allowed to. So, in a way, it may be much harder for a robot to learn its body schema than for a human.

7. Conclusion

In this paper, we have provided a general model of the body schema for humanoid robots, based on a hierarchy of FOR transformations reflecting the body structure. This model integrates the visual and proprioceptive modalities and allows a robot to form a coherent image of itself. Moreover, we provided a way to adapt this image as a result of sensory experience. This is done in a completely online and self-supervised manner so that the system can continuously adapt itself, as is the case for humans, who can adapt to visual shifts, tool use and so on. The effectiveness of the model has been shown in simulations with a 24 DOF humanoid robot and in a real setting for tool adaptation.

The results obtained suggest that the kinematic function (and for that matter its inverse) of the robot should not be dissociated from its sensory abilities, as it precisely provides the relationship between multiple sensory informations. Learning this relationship allows a permanent recalibration of the sensors and is likely to contribute to the precision, autonomy and adaptability of humanoid robots.

Acknowledgments

This work was funded by the Swiss National Science Foundation and EU Project IST-2004-004370 RobotCub.

Appendix A. Proof of Theorem 1

We consider Eqs. (3) and (4) as a continuous time dynamical system in the parameter space defined by \mathbf{l} and \mathbf{a} . Indeed, the learning step ϵ can be interpreted as the integration constant of the dynamical system, in which case $\epsilon^{-1}\Delta\mathbf{l}$ and $\epsilon^{-1}\Delta\mathbf{a}$ tend respectively to $\frac{\partial}{\partial t}\mathbf{l}$ and $\frac{\partial}{\partial t}\mathbf{a}$, where t denotes the time.

In order to simplify the notation, we put the two parameter vectors \mathbf{l} and \mathbf{a} into a single parameter vector, $\mathbf{p} = [\mathbf{l}^T \mathbf{a}^T]^T$. We then consider the function $\mathbf{E}(\mathbf{p})$, defined by:

$$\mathbf{E}(\mathbf{p}) = \langle \mathbf{E}^\theta(\mathbf{p}) \rangle, \quad \text{with } \mathbf{E}^\theta(p) = \frac{1}{2} \|\mathcal{T}^*(\mathbf{v}_n) - \mathcal{T}_{\mathbf{p}}(\mathbf{v}_n)\|^2, \quad (\text{A.1})$$

where $\mathcal{T}_{\mathbf{p}}$ is the estimated rigid-body transformation parametrized by \mathbf{p} and dependent on the joint configuration θ [$\mathcal{T}_{\mathbf{p}} = \mathbf{l} + \mathbf{R}_{\mathbf{a}}^\theta(\mathbf{v}_n)$], $\mathcal{T}^*(\mathbf{v}_n) = \mathbf{l}^* + \mathbf{R}_{\mathbf{a}^*}^\theta(\mathbf{v}_n)$ is the real underlying transformation, and $\langle \cdot \rangle$ denotes the expectation operator assuming that θ follows symmetric pdf centered on 0. Since grad is a linear operator, $\langle [\frac{\partial}{\partial t}\mathbf{l} \ \frac{\partial}{\partial t}\mathbf{a}] \rangle = -\text{grad}\mathbf{E}(\mathbf{p})$. So (3) and (4) correspond to a gradient descent on \mathbf{E} .

It remains to be shown that if $\frac{\partial}{\partial t}\mathbf{E}(\mathbf{p}) = 0$, then $\mathcal{T}_{\mathbf{p}} = \mathcal{T}^*$, which amounts to saying that there are no local minima to \mathbf{E} . To this end we compute the gradient of \mathbf{E} :

$$\text{grad}_{\mathbf{p}}\mathbf{E} = \left\langle (\mathcal{T}_{\mathbf{p}}(\mathbf{v}) - \mathcal{T}^*(\mathbf{v}))^T \frac{\partial}{\partial \mathbf{p}} \mathcal{T}_{\mathbf{p}} \right\rangle \quad (\text{A.2})$$

$$= \langle (\mathcal{T}_{\mathbf{p}}(\mathbf{v}) - \mathcal{T}^*(\mathbf{v}))^T [\mathbf{I} \ \bar{\mathbf{R}}_{\mathbf{a}}^\theta(\mathbf{v})] \rangle. \quad (\text{A.3})$$

We can divide the gradient vector into its two components, $\text{grad}_{\mathbf{p}}\mathbf{E} = [\text{grad}_{\mathbf{l}}\mathbf{E} \ \text{grad}_{\mathbf{a}}\mathbf{E}]$. We then have

$$\text{grad}_{\mathbf{l}}\mathbf{E} = \langle (\sin\theta)(\mathbf{a} - \mathbf{a}^*) \times \mathbf{v} + \langle 1 - \cos\theta \rangle (\mathbf{v}^T \mathbf{a} \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^*) + \mathbf{l} - \mathbf{l}^* \rangle^T \quad (\text{A.4})$$

$$\begin{aligned} \text{grad}_{\mathbf{a}}\mathbf{E} = & \langle (\sin\theta(\mathbf{a} - \mathbf{a}^*) \times \mathbf{v} + (1 - \cos\theta)(\mathbf{v}^T \mathbf{a} \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^*) + \mathbf{l} - \mathbf{l}^*)^T \\ & (\sin\theta \mathbf{v} \uparrow + (1 - \cos\theta)(\mathbf{a} \mathbf{v}^T + (\mathbf{a}^T \mathbf{v}) \mathbf{I})) \rangle. \end{aligned} \quad (\text{A.5})$$

Developing this product leads to the following sum:

$$\begin{aligned} \text{grad}_{\mathbf{a}}\mathbf{E} = & \langle \sin^2\theta \mathbf{v} \times ((\mathbf{a} - \mathbf{a}^*) \times \mathbf{v}) + \langle (1 - \cos\theta) \sin\theta \rangle ((\mathbf{v} \times (\mathbf{a} - \mathbf{a}^*))^T \\ & (\mathbf{a} \mathbf{v}^T + \mathbf{a}^T \mathbf{v} \mathbf{I}) + \langle \sin\theta(1 - \cos\theta) \rangle (\mathbf{v} \times (\mathbf{v}^T \mathbf{a} \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^*)) \\ & + \langle (1 - \cos\theta)^2 \rangle (\mathbf{v}^T \mathbf{a} \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^*)^T (\mathbf{a} \mathbf{v}^T + \mathbf{a}^T \mathbf{v} \mathbf{I}) \\ & + (\mathbf{l} - \mathbf{l}^*)^T (\langle \sin\theta \rangle \mathbf{v} \uparrow + \langle 1 - \cos\theta \rangle (\mathbf{a} \mathbf{v}^T + \mathbf{a}^T \mathbf{v} \mathbf{I})) \rangle. \end{aligned} \quad (\text{A.6})$$

As the θ are zero-centered and symmetric, we have $\langle \sin\theta \rangle = \langle \sin(\theta)(1 - \cos\theta) \rangle = 0$. Thus, we have

$$\text{grad}_{\mathbf{l}}\mathbf{E} = \langle (1 - \cos\theta)(\mathbf{v}^T \mathbf{a} \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^*) + \mathbf{l} - \mathbf{l}^* \rangle^T \quad (\text{A.7})$$

$$\begin{aligned} \text{grad}_{\mathbf{a}}\mathbf{E} = & \langle \sin^2\theta \rangle (\|\mathbf{v}\|^2 (\mathbf{a} - \mathbf{a}^*) - \mathbf{v}^T (\mathbf{a} - \mathbf{a}^*) \mathbf{v}) \\ & + \langle (1 - \cos\theta)^2 \rangle (\mathbf{v}^T \mathbf{a} \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^*)^T (\mathbf{a} \mathbf{v}^T + \mathbf{a}^T \mathbf{v} \mathbf{I}) \\ & + \langle 1 - \cos\theta \rangle (\mathbf{l} - \mathbf{l}^*)^T (\mathbf{a} \mathbf{v}^T + \mathbf{a}^T \mathbf{v} \mathbf{I}). \end{aligned} \quad (\text{A.8})$$

Now, \mathbf{p} is a fixed point of the dynamical system if and only $\text{grad}_{\mathbf{l}}\mathbf{E} = \mathbf{0}$ and $\text{grad}_{\mathbf{a}}\mathbf{E}$ is colinear to \mathbf{a} (thus taking the normalization of \mathbf{a} into account). Setting $\text{grad}_{\mathbf{l}}\mathbf{E}$ to zero, we obtain

$$\mathbf{l} - \mathbf{l}^* = -\langle 1 - \cos \theta \rangle (\mathbf{v}^T \mathbf{a} \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^*) \quad (\text{A.9})$$

$$\begin{aligned} \text{grad}_{\mathbf{a}}\mathbf{E} &= \langle \sin^2 \theta \rangle (\|\mathbf{v}\|^2 (\mathbf{a} - \mathbf{a}^*) - \mathbf{v}^T (\mathbf{a} - \mathbf{a}^*) \mathbf{v})^T + (\langle (1 - \cos \theta)^2 \rangle \\ &\quad - \langle (1 - \cos \theta) \rangle^2) (\mathbf{v}^T \mathbf{a} \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^*)^T (\mathbf{a} \mathbf{v}^T + \mathbf{a}^T \mathbf{v} \mathbf{I}) \end{aligned} \quad (\text{A.10})$$

$$\begin{aligned} &= \langle \sin^2 \theta \rangle (\|\mathbf{v}\|^2 (\mathbf{a} - \mathbf{a}^*) - \mathbf{v}^T (\mathbf{a} - \mathbf{a}^*) \mathbf{v})^T \\ &\quad + \text{var}(\cos \theta) (\mathbf{v}^T \mathbf{a} \mathbf{v} + (\mathbf{v}^T \mathbf{a})^2 \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^T \mathbf{a}^* \mathbf{v} - \mathbf{v}^T \mathbf{a}^* \mathbf{v}^T \mathbf{a} \mathbf{a}^*)^T. \end{aligned} \quad (\text{A.11})$$

With no loss of generality and to simplify the notation, we can drop $\|\mathbf{v}\|^2$ and consider \mathbf{v} to be of unit norm. To further lighten the notation, we define $c = \text{var}(\cos \theta)$ and $s = \langle \sin^2 \theta \rangle = \text{var}(\sin \theta)$. We have a fixed point if and only if $\text{grad}_{\mathbf{a}}\mathbf{E}$ is colinear to \mathbf{a} or, equivalently, if it is perpendicular to the projection of any vector \mathbf{r} on the plane orthogonal to \mathbf{a} , $(\mathbf{I} - \mathbf{a} \mathbf{a}^T) \mathbf{r}$. Hence we have

$$\mathbf{0} = \text{grad}_{\mathbf{a}}\mathbf{E} \cdot (\mathbf{I} - \mathbf{a} \mathbf{a}^T) \quad (\text{A.12})$$

$$\begin{aligned} &= s(-\mathbf{a}^* + \mathbf{a}^T \mathbf{a}^* \mathbf{a} - \mathbf{v}^T \mathbf{a} \mathbf{v} + \mathbf{v}^T \mathbf{a}^* \mathbf{v} + (\mathbf{v}^T \mathbf{a})^2 \mathbf{a} - \mathbf{v}^T \mathbf{a} \mathbf{v}^T \mathbf{a}^* \mathbf{a}) \\ &\quad + c(\mathbf{v}^T \mathbf{a} \mathbf{v} - (\mathbf{v}^T \mathbf{a})^2 \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^T \mathbf{a}^* \mathbf{v} + \mathbf{v}^T \mathbf{a} \mathbf{v}^T \mathbf{a}^* \mathbf{a}^T \mathbf{a}^* \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{v}^T \mathbf{a} \mathbf{a}^* \\ &\quad + \mathbf{v}^T \mathbf{a}^* \mathbf{v}^T \mathbf{a} \mathbf{a}^T \mathbf{a}^* \mathbf{a}) \end{aligned} \quad (\text{A.13})$$

$$\begin{aligned} &= s(\mathbf{v}^T \mathbf{a}^* - \mathbf{v}^T \mathbf{a})(\mathbf{v} - \mathbf{a}^T \mathbf{v} \mathbf{a}) - \mathbf{a}^* + \mathbf{a}^T \mathbf{a}^* \mathbf{a} \\ &\quad + c(\mathbf{v}^T \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^T \mathbf{a}^*)(\mathbf{v} - \mathbf{a}^T \mathbf{v} \mathbf{a}) - \mathbf{v}^T \mathbf{a} \mathbf{v}^T \mathbf{a}^*(\mathbf{a}^* - \mathbf{a}^T \mathbf{a}^* \mathbf{a}) \end{aligned} \quad (\text{A.14})$$

$$\begin{aligned} &= (s(\mathbf{v}^T \mathbf{a}^* - \mathbf{v}^T \mathbf{a}) + c(\mathbf{v}^T \mathbf{a} - \mathbf{v}^T \mathbf{a}^* \mathbf{a}^T \mathbf{a}^*)) (\mathbf{v} - \mathbf{a}^T \mathbf{v} \mathbf{a}) \\ &\quad - (c \mathbf{v}^T \mathbf{a} \mathbf{v}^T \mathbf{a}^* + s)(\mathbf{a}^* - \mathbf{a}^T \mathbf{a}^* \mathbf{a}). \end{aligned} \quad (\text{A.15})$$

We now notice that this last expression is the weighted sum of two vectors, $\mathbf{v} - \mathbf{a}^T \mathbf{v} \mathbf{a}$ and $\mathbf{a}^* - \mathbf{a}^T \mathbf{a}^* \mathbf{a}$. These are respectively the projections of \mathbf{v} and \mathbf{a}^* on the plane orthogonal to \mathbf{a} . This implies that \mathbf{a} , \mathbf{v} , \mathbf{a}^* are coplanar and we can rewrite (A.15) as

$$0 = (s(\cos \alpha^* - \cos \alpha) + c(\cos \alpha - \cos \alpha^* \cos \gamma)) \sin \alpha - (s + c \cos \alpha \cos \alpha^*) \sin \gamma, \quad (\text{A.16})$$

where α is the angle between \mathbf{a} and \mathbf{v} , α^* is the angle between \mathbf{a}^* and \mathbf{v} , and γ is the angle between \mathbf{a} and \mathbf{a}^* , as depicted in Fig. 14. Since $\gamma = \alpha^* - \alpha$, we have

$$\sin \gamma = \sin \alpha^* \cos \alpha - \cos \alpha^* \sin \alpha \quad \cos \gamma = \cos \alpha \cos \alpha^* + \sin \alpha \sin \alpha^*. \quad (\text{A.17})$$

Inserting (A.17) into (A.16) yields

$$\begin{aligned} 0 &= (s(\cos \alpha^* - \cos \alpha) + c(\cos \alpha - \cos \alpha^* (\cos \alpha \cos \alpha^* + \sin \alpha \sin \alpha^*))) \sin \alpha \\ &\quad - (s + c \cos \alpha \cos \alpha^*) (\sin \alpha^* \cos \alpha - \cos \alpha^* \sin \alpha) \end{aligned} \quad (\text{A.18})$$

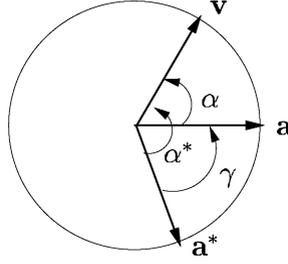


Fig. 14. Going from (A.15) to (A.16). All of the three fixed points \mathbf{a}^* , \mathbf{a}_1 and \mathbf{a}_2 are located on the plane defined by \mathbf{a}^* and \mathbf{v} .

$$= (c - s) \cos \alpha \sin \alpha + 2s \cos \alpha^* \sin \alpha - s \sin \alpha^* \cos \alpha + c(-\cos^2 \alpha^* \cos \alpha \sin \alpha - \cos \alpha^* \sin \alpha^* \sin^2 \alpha - \cos \alpha^* \sin \alpha^* \cos^2 \alpha + \cos^2 \alpha^* \cos \alpha \sin \alpha) \quad (\text{A.19})$$

$$= (c - s) \cos \alpha \sin \alpha + 2s \cos \alpha^* \sin \alpha - s \sin \alpha^* \cos \alpha - c \cos \alpha^* \sin \alpha^* \quad (\text{A.20})$$

$$= \frac{1}{2}(c - s) \sin(2\alpha) + s \sin(\alpha - \alpha^*) + s \cos \alpha^* \sin \alpha - \frac{c}{2} \sin(2\alpha^*) \quad (\text{A.21})$$

$$= \frac{1}{2}(c - s) \sin(2\alpha) + \frac{3s}{2} \sin(\alpha - \alpha^*) + \frac{s}{2} \sin(\alpha + \alpha^*) - \frac{c}{2} \sin(2\alpha^*). \quad (\text{A.22})$$

By performing the change of variables $\beta = \alpha - \alpha^*$, $\phi = \alpha + \alpha^*$, we can rewrite this last equation as

$$0 = \frac{1}{2}(c - s) \sin(\beta + \phi) + \frac{3s}{2} \sin \beta + \frac{s}{2} \sin \phi + \frac{c}{2} \sin(\beta - \phi) \quad (\text{A.23})$$

$$= c \sin \beta \cos \phi + \frac{3s}{2} \sin \beta + \frac{s}{2} (\sin \phi - \sin(\beta + \phi)) \quad (\text{A.24})$$

$$= \left(c \cos \phi + \frac{3s}{2} \right) \sin \beta - s \sin \frac{\beta}{2} \cos \left(\frac{\beta}{2} + \phi \right) \quad (\text{A.25})$$

$$= (2c \cos \phi + 3s) \sin \frac{\beta}{2} \cos \frac{\beta}{2} - s \sin \frac{\beta}{2} \cos \left(\frac{\beta}{2} + \phi \right) \quad (\text{A.26})$$

$$= \sin \frac{\beta}{2} \left((2c \cos \phi + 3s) \cos \frac{\beta}{2} - s \cos \left(\frac{\beta}{2} + \phi \right) \right) \quad (\text{A.27})$$

$$= \sin \left(\frac{\alpha - \alpha^*}{2} \right) \left(2c \cos(\alpha + \alpha^*) \cos \left(\frac{\alpha - \alpha^*}{2} \right) + 3s \cos \left(\frac{\alpha - \alpha^*}{2} \right) - s \cos \left(\frac{3\alpha + \alpha^*}{2} \right) \right) \quad (\text{A.28})$$

$$= \sin \left(\frac{\alpha - \alpha^*}{2} \right) \left((c - s) \cos \left(\frac{3\alpha + \alpha^*}{2} \right) + 3s \cos \left(\frac{\alpha - \alpha^*}{2} \right) + c \cos \left(\frac{\alpha + 3\alpha^*}{2} \right) \right). \quad (\text{A.29})$$

The first factor indicates that, as expected, $\alpha = \alpha^*$ is a solution. The other solutions are given by the zeros of the second factor. If $|c - s| \leq |s|$, i.e. $c < 2s$, this factor has at most two zeros, so there are at most three solutions to (A.29). Since they all lie on a circle, there is one minimum, one maximum and possibly a plateau. So there is only one minimum in \mathbf{E} , which completes the proof.

References

1. T. N. Aflalo and M. S. A. Graziano, Relationship between unconstrained arm movements and single-neuron firing in the macaque motor cortex, *J. Neurosci.* **27** (2007) 2760–2780.
2. O. A. Bauchau and L. Trainelli, The vectorial parameterization of rotation, *Nonlinear Dynam.* **32** (2003) 71–92.
3. F. L. Bedford, Keeping perception accurate, *Trends Cogn. Sci.* (1999).
4. M. Botvinick, Probing the neural basis of body ownership, *Science* **305**(5685) (2004) 782–783.
5. J. W. Burdick, On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds, in *Proc. IEEE Int. Conf. Robotics and Automation* (1989), pp. 264–270.
6. Y. Burnod, P. Baraduc, A. Battaglia-Mayer, E. Guigon, E. Koechlin, S. Ferraina, F. Laquaniti and R. Caminiti, Parieto-frontal coding of reaching: An integrated framework, *Exp. Brain Res.* **129** (1999) 325–346.
7. A. D’Souza, S. Vijayakumar and S. Schaal, Learning inverse kinematics, in *Proc. 2001 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (2001), pp. 298–303.
8. L. Fadiga, L. Fogassi, V. Gallese and G. Rizzolatti, Visuomotor neurons: Ambiguity of the discharge or “motor” perception? *Int. J. Psychophysiol.* **35** (2000) 165–177.
9. S. Fuke, M. Ogino and M. Asada, Body image constructed from motor and tactile images with visual information, *Int. J. Humanoid Robot.* **4**(2) (2007) 347–364.
10. C. Gaskett and G. Cheng, Online learning of a motor map for humanoid robot reaching, in *Proc. 2nd Int. Conf. Computational Intelligence, Robotics and Autonomous Systems* (2003).
11. M. S. A. Graziano and C. G. Gross, A bimodal map of space: Somatosensory receptive fields in the macaque putamen with corresponding visual receptive fields, *Exp. Brain Res.* **97** (1993) 96–109.
12. A. Guez and Z. Ahmad, Accelerated convergence in the inverse kinematics via multi-layer feedforward networks, in *Proc. Int. Joint Conf. Neural Networks* (1989), pp. 341–344.
13. T. Higuchi, K. Imanaka and A. E. Patla, Action-oriented representation of peripersonal and extrapersonal space: Insights from manual and locomotor action, *Jpn. Psychol. Res.* (2006).
14. N. P. Holmes and C. Spence, The body schema and multisensory representation(s) of peripersonal space, *Cogn. Process* (2004).
15. H. Imamizu, Y. Uno and M. Kawato, Adaptive internal model of intrinsic kinematics involved in learning an aiming task, *J. Exp. Psychol. Human Perception Performance* **24** (1998) 812–829.
16. T. Jellema, G. Maassen and D. I. Perrett, Single cell integration and aminate form, motion and location in the superior temporal cortex of the macaque monkey, *Cerebral Cortex* **14** (2004) 781–790.

17. N. Kumar and L. Behera, Visual–motor coordination using a quantum clustering based neural control scheme, *Neural Process. Lett.* **20**(1) (2004) 11–22.
18. M. Kuperstein, Neural model for adaptive hand–eye coordination for single postures, *Science* (1988).
19. E. Làdavas, Functional and dynamical properties of visual peripersonal space, *Trends Cogn. Sci.* (2002).
20. M. Lopes and J. Santos-Victor, Visual learning by imitation with motor representations, *IEEE Trans. Syst. Man Cybern. Part B* (2005).
21. A. Maravita and A. Iriki, Tools for the body (schema), *Trends Cogn. Sci.* **8**(2) (2004).
22. G. Metta, G. Sandini and J. Konczak. A developmental approach to visually-guided reaching in artificial systems, *Neural Networks* **12**(10) (1999) 1413–1427.
23. Y. Nakamura and H. Hanafusa, Inverse kinematics solutions with singularity robustness for robot manipulator control, *ASME J. Dynam. Syst. Meas. Contr.* **108** (1986) 163–171.
24. J. K. O’Regan and A. Noë, A sensorimotor account of vision and visual consciousness, *Behav. Brain Sci.* **24**(5) (2001) 939–1011.
25. A. Pouget, S. Deneve and J. R. Duhamel, A computational perspective on the neural basis of multisensory spatial representations, *Nat. Rev. Neurosci.* **3** (2002) 741–747.
26. M. J. Prud’homme and J. F. Kalaska, Proprioceptive activity in primate primary somatosensory cortex during active arm reaching movements, *J. Neurophysiol.* **72**(5) (1994) 2280–2301.
27. G. Rizzolatti, L. Fadiga, L. Fogassi and V. Gallese, The space around us, *Science* (1997).
28. G. Sandini, G. Metta and D. Vernon, Robotcub: An open framework for research in embodied cognition, in *Proc. 2004 IEEE/RAS Int. Conf. Humanoid Robotics* (2004), pp. 13–32.
29. L. Sciavicco and B. Siciliano, A solution algorithm to the inverse kinematic problem for redundant manipulators, *IEEE J. Robot. Autom.* (1988).
30. S. I. Tillery, J. F. Soechting and T. J. Ebner, Somatosensory cortical activity in relation to arm posture: Nonuniform spatial tuning, *J. Neurophysiol.* **76** (1996) 2423–2438.
31. J. A. Walter and K. J. Schulten, Implementation of self-organizing neural networks for visuo-motor control of an industrial robot, *IEEE Trans. Neural Networks* **4**(1) (1993) 86–95.
32. C. W. Wampler, Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods, *IEEE Trans. Syst. Man Cybern. Part C* **16**(1) (1986) 93–101.
33. R. B. Welch, Adaptation of space perception, in *Handbook of Perception and Human Performance*, eds. K. R. Boff, L. Kaufman and J. P. Thomas (1986), pp. 24:1–24:45.
34. D. E. Whitney, Resolved motion rate control of manipulators and human prostheses, *IEEE Trans. Man Mach. Syst.* **10**(2) (1969) 47–52.