Article

Adaptive Behavior

Adaptive Behavior
1–16

DOI: 10.1177/1059712319855836
journals.sagepub.com/home/adb
SAGE

From motor to visually guided bimanual affordance learning

Martí Sánchez-Fibla[1], Sébastien Forestier[2], Clément Moulin-Frier[1], Jordi-Ysard Puigbò[3] and Paul FMJ Verschure[3,4]

Abstract

The mechanisms of how the brain orchestrates multi-limb joint action have yet to be elucidated and few computational sensorimotor (SM) learning approaches have dealt with the problem of acquiring bimanual affordances. We propose a series of bidirectional (forward/inverse) SM maps and its associated learning processes that generalize from uni- to bimanual interaction (and affordances) naturally, reinforcing the motor equivalence property. The SM maps range from a SM nature to a solely sensory one: full body control, delta SM control (through small action changes), delta sensory co-variation (how body-related perceptual cues covariate with object-related ones). We make several contributions on how these SM maps are learned: (1) *Context* and *Behavior-Based Babbling*: generalizing goal babbling to the interleaving of absolute and local goals including guidance of reflexive behaviors; (2) *Event-Based Learning*: learning steps are driven by visual, haptic events; and (3) *Affordance Gradients*: the vectorial field gradients in which an object can be manipulated. Our modeling of bimanual affordances is in line with current robotic research in forward visuomotor mappings and visual servoing, enforces the motor equivalence property, and is also consistent with neurophysiological findings like the multiplicative encoding scheme.

Keywords

Sensorimotor learning, affordances, goal babbling, motor equivalence, interlimb coordination, bimanual affordances

Handling Editor: Philipp Zech, University of Innsbruck, Austria

1. Introduction

The mechanisms of how the brain orchestrates multi-limb joint action to interact with a dynamic environment have yet to be elucidated (Takiyama & Sakai, 2016). There is evidence by Corbetta and Thelen (1996) that bimanual interactions precede unimanual during development: the newborn starts by using synchronous bimanual control movements (of 1 degree of freedom (DOF)) which may have large perceptual sensory effects: visual (i.e. several visual perceptual cues reducing in scale and rotating synchronously after a box has been hit) and haptic (hundreds of tactile sensor activations). Affordances, after Gibson (1986), are the categorizations of such goal-relevant properties of objects. The learning substrate of multi-limb affordances has not been well addressed in the computational sensorimotor (SM) and affordance learning (AL) research: they are only mentioned in Zech et al. (2017) suggesting to be derived from "chaining" or combination of other simpler affordances and do not appear in Thill, Caligiore, Borghi, Ziemke, and Baldassarre (2013). Early synchronized bimanual movements can also enable "pull-toward" affordances very quickly, impossible to achieve otherwise. These facts raise interesting questions. (a) How we start making sense of sensory covariations given this simple initial control? It would seem like a mechanism since birth must be in place (early attention to biological motion in the newborn, as reported by Simion, Regolin, & Bulf, 2008, could be a proof of that). (b) May sensory covariations be learned

[1]Pompeu Fabra University, Barcelona, Spain
[2]Flowers Team, Inria Bordeaux, Talence, France
[3]Synthetic Perceptive Emotive Cognitive Systems, Institute for Bioengineering of Catalonia (IBEC), Barcelona, Spain
[4]Catalan Institution for Research and Advanced Studies (ICREA), Barcelona, Spain

Corresponding author:
Martí Sánchez-Fibla, Pompeu Fabra University, Roc Boronat 138, 08018 Barcelona, Spain.
Email: marti.sanchez@upf.edu

separately? This would certainly benefit the transfer of learning from bi- to unimanual interactions and would resolve Bernstein's "motor equivalence problem" dealing with an increasing motor program repertoire during development (Sporns & Edelman, 1993). (c) How we drive learning from large changes in the sensory space? Learning needs to be driven by small action changes, in the motor space (see Bullock, Grossberg, & Guenther, 1993), what we call delta actions (Braud, Pitti, & Gaussier, 2018, and, Escobar-Juárez, Schillaci, Hermosillo-Valadez, & Lara-Guzmán, 2016, consider learning of affordances through delta actions), and also in the sensory space (visual servoing approaches that learn the so-called visual Jacobian; Hosoda & Asada, 1994).

We define and implement the learning process (based on delta actions) of a set of four mappings (two of a solely sensory nature) of increasing complexity to address the distinction between motor and sensory affordances, motor and visually guided AL, uni- and bimanual affordances:

1. *Forward/inverse full body control*. Mappings from the absolute motor state to the absolute sensory state.
2. *Forward/inverse delta control*. Mapping small delta motor actions and how these affect body movement perceptual cues.
3. *Singular interaction control*. Mappings that consider how single body-related perceptual cues covariate with other sensory movement cues (i.e. from the object).
4. *Multiple interaction control (bimanual for short)*. Mappings that consider how two or more body-related perceptual cues covariate.

Map types 1 and 2 are of a sensory motor nature and types 3 and 4 are what we mean by sensory mappings (types 3 and 4) is that they implement predictions of sensory effects and its consequences on the sensory side (as we do in Maffei, Herreros, Sanchez-Fibla, Friston, & Verschure, 2017) and contain no direct motor information. Maps consider delta actions and delta sensory changes (the consequences of small changes in either motor or visual domains). The motor equivalence property is enforced as the same sensory delta effect can be performed with many motor programs (by querying mapping 2). The consideration of delta sensory and motor changes oblige us to introduce context: one can query a mapping while being in a particular context (i.e. object position, absolute motor state). Sensory mappings are affected as well by the body as learning is guided by exploration and what can be explored depends on the body. For very redundant spaces, goal babbling (GB) driven exploration has been proven efficient (Rolf, Steil, & Gienger, 2010) and consistent with developmental approaches (Rolf & Asada, 2014) because it generates goals in the sensory space and avoids exploring redundant motor goals. Delta mappings call for a reformulation of GB in which we interleave the generation of absolute and delta goals (goals relative to the current absolute state): we present this method as algorithm *Context Goal Babbling*: it learns from sampling delta goals within a context. Many generated goals are still far from reaching contact with the object. We define *Behavior-Based Babbling,* which adds to *Context Goal Babbling* guidance through reactive behaviors which trigger object attraction and implements a sensory GB learning. Learning can be triggered by any change in the sensory domain as presented in *Event-Based Learning* algorithm. From mappings 3 and 4, visual affordances can be defined that we call affordance gradients (AGs), body-independent projections in the sensory space of the manipulation abilities of objects. AGs can depend on single or multiple points of actuation. The so-called bimanual mapping is used to learn bimanual affordances (bimanual AGs) in a simulated dual arm robot[1], and it's also used to perform successfully a peg-in-hole task.

2. Methods

A motor state $m = \langle m_0, ..., m_{n-1} \rangle$ corresponds to the angles of every joint $m_i \in [0...2\pi]$. The motor space is denoted by $\mathcal{M}$. A delta motor state $\Delta m = \langle \Delta m_0, ..., \Delta m_{n-1} \rangle$ are the velocities of each joint, being n the total number of DOFs. Concretely, we use a $n = 6$ simulated robot torso with two arms (see Note 1) and an object (a square, a circle, or a triangle) shown in Figure 1(a). Each arm has symmetric joint limits:

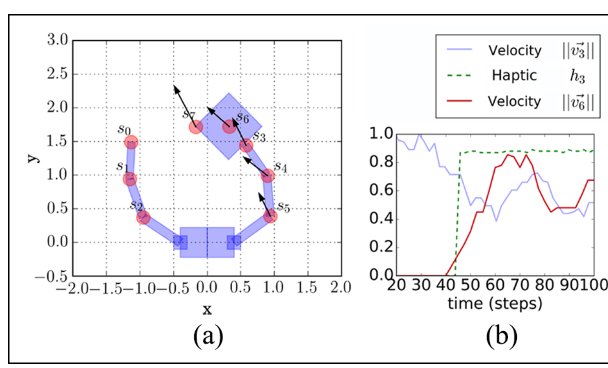

Figure 1. Robotic setup. (a) A simulated two-arm fixed torso, being $s_0, ..., s_2$ the sensory interest points (SIPs) in the left arm, $s_3, ..., s_5$ the ones in the right arm, and s_6, s_7 the ones in the object. Simulation and results are available (see Note 1). The velocity vectors are superimposed. (b) Signals from the SIPs in (a): the velocity of SIPs s_3 (the end-effector) and s_6 (center of the object) are plotted together with the activation of the haptic signal of s_3. The y axis is normalized from 0 to 1, so units are arbitrary.

$m_0 \in [-0.9\pi...0.2\pi]$, $m_1 \in [-0.9\pi...0]$, and $m_2 \in [-0.4\pi...0.4\pi]$ for the left arm and $m_3 \in [0.2\pi... -0.9\pi]$, $m_4 \in [0... -0.9\pi]$, and $m_5 \in [-0.4\pi... 0.4\pi]$ for the right one.

The sensory space (denoted by $\mathcal{S}$) includes two modalities, visual motion information and haptic inputs, both gathered in what we define as sensory interest point (SIP). A SIP is attached at a relevant feature of the visual field (i.e. the end-point effector, at the junction between two arm links, at the corner of a square object) and could be computed as SIFT features defined by Lowe (1999) from the scene image (see Figure 1(a)). A sensory state s consists of a set of SIPs: $s = \{s_0, \ldots, s_i, \ldots\}$ each being a tuple

$$s_i = \langle p_i = \langle x_i, y_i \rangle, h_i \rangle$$

where p_i corresponds to its two-dimensional (2D) Cartesian coordinates and h_i a real value haptic signal normalized from 0 to 1. From the temporal SM data, one can extract information about the changing signals for each SIP

$$\Delta s_i = \langle \vec{v}_i = \langle v_i^x, v_i^y \rangle, \Delta h_i \rangle$$

where $\vec{v}_i$ denotes the velocity vector of s_i, $||\vec{v}_i|| \in [0..1]$ its magnitude, and Δh_i the haptic signal change. We call Δs the set of delta signals of each SIP s_i: $\Delta s = \{\Delta s_0, \ldots, \Delta s_i, \ldots\}$. In total, eight SIPs are considered: three in each arm of the robot. End effector SIPs will be denoted by $s_{left} = s_0$ and $s_{right} = s_3$. Two SIPs are added at the object: $s_{obj} = s_6$ at the center and s_7 in one corner (for the box and triangle) or lateral (circle) (see Figures 1(a) and 6).

2.1. SM mappings

We identify a series of SM mappings of increasing complexity that separate motor from visually guided action. The mappings are built on a different interpretation of "action" itself. Action, at its lower level, is the group of motor signals sent to the muscles or motors (as many signals as DOFs). At a higher level, we can also interpret action as the result of its sensory consequences: a change at a sensory level, a movement of a perceptual cue (the so-called SIPs, sensory points s_i) in the visual modality. Changes at the sensory level (we refer to it as Δs_i) can also be interpreted as actions: some are caused by low-level motor actions (arm motors causes related perceptual cues to move in the visual field), others are indirectly caused (movement of object-related cues due to contact with the arm), and others may be caused even more indirectly by, for example, gravity. Dependencies and consequences in the sensory modality are captured by learning the mutual covariations of SIPs. SIPs have been inspired by the biological motion phenomena in which humans can perceive and understand complex action having access to some few points linked to relevant positions of a human pose (Simion et al., 2008).

We define now, the following mappings:

2.1.1. Forward/inverse full body mapping. There is a one-to-one mapping from m (body state) to s (perceptual state)

$$f_i^{abs}(m) = s_i \tag{1}$$

where f_i^{abs} is the absolute mapping of every SIP s_i. In its forward form, given the motor state m, returns its position p_i. Its inverse form returns the motor state to be achieved given the position p_i of the SIP. The inverse mapping is not uniquely defined as there may exist many motor states achieving a single sensory position. We will denote the inverse of all mappings f^{-1} by g. The inverse map is then $g_i^{abs}(s_i) = m$. This mapping is often studied in the SM learning and GB literature (Benureau, Fudal, & Oudeyer, 2014; Forestier & Oudeyer, 2016b; Moulin-Frier & Oudeyer, 2013; Rolf et al., 2010) and does not take into account time, nor small changes of actions, that we call delta actions.

2.1.2. Delta motor to sensory. Delta mappings capture the effects of applying delta changes in motor (Δm) and sensory (Δs) space. They subsume the absolute mapping $f_i^{abs}(m) = s$, but as we will see, they are useful when incorporating the notion of context. We may want to know what will be the result of a delta action knowing the current state, and we may want to know the delta action to apply if we want to observe a delta sensory change. The delta motor to sensory mapping takes as input a motor state m and a delta motor command Δm and returns the resulting sensory state s_i (including position p_i) and velocity $\vec{v}_i = \Delta s_i$ (m acts as context here)

$$f_i^{\Delta m}(m, \Delta m) = \langle m, s_i, \Delta s_i \rangle \tag{2}$$

where m and s_i are the motor and SIP states before a motor variation Δm was applied and a sensory effect Δs_i was observed. The state m acts as context here and fully determines any s_i. This mapping subsumes $f_i^{abs}(m) = s_i$.

Forward/inverse delta control mappings are reminiscent of the work of Bullock et al. (1993), have been considered to learn affordances in Braud et al. (2018), and relate to visual servoing and the visual Jacobian in robotics (see, for example, Hosoda & Asada, 1994, in which the so-called visual Jacobian considering visuomotor variations is learned). Only through mapping (2), a desired perceptual change can be provoked by querying the appropriate motor delta command to be performed. Mappings (3) and (4) are purely sensory in the sense that they capture how sensory perceptual cues (SIPs) covariate.

2.1.3. Two interacting SIPs. A sensory mapping for every two SIPs

$$f_{ij}^{\Delta s}(s_j, \Delta s_j, s_i) = \langle s_i, \Delta s_i \rangle \tag{3}$$

Given the position and velocity of s_j together with another SIP s_i, the mapping returns the resulting velocity Δs_i and its position as a context. Intuitively, this mapping captures the fact that if s_i is moving or it stopped, the cause may be s_j. An s_i may be a SIP on the object (denoted by s_{obj}). If s_{obj} is moving in a certain direction $\vec{v}_{obj}$, a possible cause is s_j being in contact with the object and pushing it toward $\vec{v}_i$.

This mapping is completely sensory and bringing the mappings to the sensory side when possible makes them independent of the body. If we make one of the segments of one arm longer, the motor mappings have to be relearned but not the sensory ones. This continues to be the case if we include another segment in the arm. Another important advantage is the scalability of the mapping. If we are considering a very high dimensional system like the human arm (including the hand with its about 30 DOFs), learning the mappings that include motor components would be much costly (3 compared to 30 input DOFs).

The sensory mappings have to work in conjunction with the previous learned delta mappings if we want to reach bidirectionally. For instance, if we want to know the delta motor command that we need to achieve a velocity in a certain position of the SIP p_i, we can access the inverse mapping: $g_i^{\Delta m}(m, s_i, \Delta s_i) = \Delta m$, with the motor state m as context. We cannot use s_i here as the motor state for a given s_i is ambiguous. If we need to know the absolute motor command that can bring us to position p_i, we can use again: $g_i^{abs}(s_i) = m$.

2.1.4. Three interacting SIPs. For every two SIPs, we consider an SM mapping to be able to address bimanual affordances. If one SIP s_i moves, every two other SIPs are selected as possible causes of that movement

$$f_{ijk}^{\Delta s}(s_j, s_k, \Delta s_j, \Delta s_k, s_i) = \langle s_i, \Delta s_i \rangle \tag{4}$$

This mapping will allow us to capture bimanual affordances when considering the target SIPs on the object: $s_i = s_{obj}$ with $i \in \{6, 7\}$ in our case.

2.2. Exploration strategies

In a robotic context, the aforementioned mappings have to be learned through the interaction of the agent with the environment. This is a departure from classical machine learning approaches where the training set is generally provided by some external process (e.g. a given database of images). In our case, the training set is iteratively built according to the robot activity, generating data about the effects of executed motor actions. This has two important consequences. First, the learning has to occur *online*, that is, the learned SM mappings have to be updated regularly according to the robot experience. Second, the learning is considered *active*, that is, the robot has to actively choose how to explore the world in order to iteratively generate an informative training set for the considered mappings and how they will be used to solve the required task (e.g. to perform forward or inverse predictions).

Moreover, the learning of the mappings is not guided by reward as reinforcement learning (RL) methods are (see section 4.5). The problem we are addressing is a typical supervised learning problem: given a set of observed (t_{in}, t_{out}) tuples, how to predict t_{out} given an unseen t_{in} (forward prediction), and how to infer t_{in} given an unseen target t_{out} (inverse prediction). We use the general notation (t_{in}, t_{out}) and not motor $(m \in \mathcal{M})$/ sensory $(s \in \mathcal{S})$ because we go beyond SM maps and sometimes they are solely sensory related. For this reason, we learn SM and sensory maps based on a supervised methodology of input–output samples. The maps can then be queried in a forward and inverse way, to know the actions to make to achieve a desired goal state.

2.3. Mappings implementation

The mappings are implemented using Explauto library (Moulin-Frier, Rouanet, & Oudeyer, 2014), which uses kd-trees to store the input and output spaces and to efficiently find the nearest neighbor of a given tuple t_{in}. When we query the kd-tree of the input side with a vector t_{in}, the nearest neighbor of t_{in} is found in the dataset, and the corresponding t_{out} is returned. Similarly, for an inverse query. The addition of context queries only adds dimensions in the input and output spaces but the query process is implemented similarly. See Note 1 for links into the implementation.

2.4. SM learning processes

For learning, we present two variations of GB which has been proven to be very effective for high DOFs (Rolf et al., 2010): *Context Goal Babbling* and *Behavior-Based Babbling*, together with a learning based on perceptual changes that we call *Event-Based Learning*.

2.4.1. GB with context. We use an active GB strategy based on the maximization of the learning progress as implemented in the Explauto library (Moulin-Frier et al., 2014): goals are generated according to where we progress more (Moulin-Frier & Oudeyer, 2013).

In this library, an interest model is a way of self-generating goals, which we denote by I in Algorithm 1. We use the "discretized_progress" interest model (available in the Explauto library). This algorithm discretizes

Algorithm 1. ContextGoalBabbling()

while $error > th_{error}$ **do**
  if random () $> th_{abs}$ **then**
1    $s^{goal} \leftarrow$ sample_goal (I^f)
2    $m \leftarrow g^{abs}(s^{goal})$
    goto_motors (m)
3 $\langle m, s \rangle \leftarrow$ get_state ()
4 $\Delta s^{goal} \leftarrow$ sample_delta_goal (I^f, s)
5 $\Delta m \leftarrow g^{\Delta m}(m, s, \Delta s^{goal})$
6 $\Delta s^{obs} \leftarrow$ delta_motors (Δm)
7 update ($f^{\Delta m}, \langle m, \Delta m \rangle, \langle s, \Delta s \rangle$)
8 update ($I^f, \langle s, \Delta s^{goal} \rangle$)
  $error \leftarrow$ evaluate_mse ($ntrials, f, g$)

Algorithm 2. EventBasedLearning()

for $s_i \in S \wedge change(\vec{v}_i)$ **do**
  $C \leftarrow \{s_j | j \neq i \wedge (h_j > 0 \vee \| \vec{v}_j \| > 0)\}$
  for $s_j, s_k \in C$ **do**
    $error \leftarrow ||\langle s_i, \Delta s_i \rangle - f_{ij}^{\Delta s}(s_j, \Delta s_j, s_i)||$
    $error_2 \leftarrow ||\langle s_i, \Delta s_i \rangle - f_{ijk}^{\Delta s}(s_j, s_k, \Delta s_j, \Delta s_k, s_i)||$
9   **if** $error > th_{error}$ **then**
      update ($f_{ij}^{\Delta s}, \langle s_j, \Delta s_j, s_i \rangle, \langle s_i, \Delta s_i \rangle$)
10   **if** $error_2 > th_{error}$ **then**
      update ($f_{ijk}^{\Delta s}, \langle s_j, s_k, \Delta s_j, \Delta s_k, s_i \rangle, \langle s_i, \Delta s_i \rangle$)

the goal space into small regions and monitors the competence in reaching the goals in each region. It will then sample new goals in the regions where the competence progress is high in order to maximize the expected future progress.

As mentioned, for the delta mapping, we need to introduce context (the sensory state in which the delta action is applied). When learning an absolute mapping, one can reset the motor state at each time, trying to achieve the new goal, setting the motors to the new inverse predicted state. But one could benefit from generating goals given the current state, to try to achieve target delta velocities given the current motors position.

Algorithm 1 does GB in such conditions in which we interleave the generation of absolute and delta goals. We sample with some probability th_{abs} an absolute goal (line 1). We query the absolute motor state to achieve it via an inverse absolute mapping (line 2). We then access the current global state (line 3) and sample a delta goal using it as context (line 4). We proceed by querying the inverse delta SM map to try to achieve the corresponding delta goal (line 5). We then execute the delta action (line 16). Then the delta sensory motor map (line 17) and the interest models are updated (line 18). Finally, a mean squared error (MSE) evaluation is performed by generating a number of trials and evaluating both forward and inverse mappings (we do not provide the details here).

To evaluate the inverse performance, we have to generate a goal that is reachable in a uniform way. In any other case, the error would be large always. In line 1, we sample a goal from the sensory space using the reachability maps generated so far (Figure 3). These maps are generated with Gaussian kernels reachable positions, thus it will generate goals that are feasible but not necessarily in the database.

2.5. An event-based learning process

An event is considered to be a detection of a change in speed (increase or decrease) of any SIP denoted $change(\vec{v}_i)$ in Algorithm 2 or any change in its associated haptic signal. A moving event of SIP s_i can then have an associated causal context involving SIPs that are moving or that have an haptic activation at that moment: $C = \{s_j | h_j > 0 \vee \| \vec{v}_j \| > 0\}$.

The SM memory is updated continuously, in this way, when an event related to sensory point s_i occurs, the forward and inverse prediction error of that event can be computed for mappings involving s_i and its causal context C. An event triggers the update of the mappings if these prediction errors are greater than a threshold (lines 9 and 10).

Consider the situation depicted in Figure 1(b). A change in velocity $\vec{v}_6$ of SIP s_6 will trigger an event. The causal context is $C = \{s_3, s_4, s_5, s_7\}$. Depending on the forward and inverse prediction errors, the mapping $f_{ij}^{\Delta s}$ can be updated for $i = 6$ and $j \in \{5, 4, 3, 7\}$.

The effects of an action on the orientation of an object can be derived from the mappings involving two SIPs on the object. Alternatively, one could include mappings that consider the rotation of an object as we will also do.

2.6. Reactive exploration

GB, as it includes an inverse prediction of the model, requires a preliminary phase of interaction with the environment to collect the first SM data bootstrapping the model. This is usually done by a preliminary phase of motor babbling. Instead, here we introduce a reactive exploration strategy based on the *Distributed Adaptive Control* (DAC) theory (Verschure, Voegtlin, & Douglas, 2003; Verschure, Pennartz, & Pezzulo, 2014) which considers that cognition relies on the interaction of four control loops operating at different levels of abstraction. Two of these control loops, called layers, are relevant for the present study: the *reactive layer*, implementing the "out-of-the-box" behavior of a cognitive agent and supposed to be pre-wired (e.g. from evolution in a biological context, or by the programmer in a robotic one), and the *adaptive layer*, supporting SM learning mechanisms from the data collected at the reactive level. We present it in Algorithm 3 and we call it *Behavior-Based Babbling*. We add a reactive reflex during the learning of SM mappings to be able to

Algorithm 3. BehaviorBasedBabbling()

while $error > th_{error}$ **do**
  if random $() > th_{abs}$ **then**
11     $\langle s_l, s_r, s_{obj} \rangle \leftarrow$ ReactiveBehaviorLoop()
12   $\Delta s_{obj}^{goal} \leftarrow$ sample_delta_goal (I^f, s_{obj})
13   $\langle \Delta s_l, \Delta s_r \rangle \leftarrow g^{\Delta s}(s_{obj}, \Delta s_{obj}^{goal})$
  if *IndividualArmMode* **then**
14     $\Delta m \leftarrow g_l^{\Delta m}(m, s_l, \Delta s_l) \cup g_r^{\Delta m}(m, s_r, \Delta s_r)$
  else
15     $\Delta m \leftarrow g^{\Delta m}(m, s_l, s_r, \Delta s_l, \Delta s_r)$
16   $\langle \Delta s_l^{obs} \Delta s_r^{obs} \Delta s_{obj}^{obs} \rangle \leftarrow$ delta_motors (Δm)
17   update $(f^{\Delta s}, \langle s_l, s_r, \Delta s_l^{obs}, \Delta s_r^{obs}, s_{obj} \rangle, \langle s_{obj}, \Delta s_{obj}^{obs} \rangle)$
18   update $(I^f, \langle \Delta s_{obj}^{obs}, \Delta s_{obj}^{goal} \rangle)$
  $error \leftarrow$ evaluate_mse $(ntrials, f, g)$

generate relevant absolute states and guide exploration (line 11). This idea has already been implemented in several computational models based on the DAC framework (e.g. Moulin-Frier, Sanchez-Fibla, & Verschure, 2015; Puigbo et al., 2015), but was not linked to the concept of exploration strategies. The reactive reflex that we introduce attracts the end-points of the arms to the object when learning the bimanual setup. *Behavior-Based Babbling*, Algorithm 3, performs reactive behaviors that serve as exploration of global states (line 11).

Behavior-Based Babbling learns only the sensory mappings and assumes the SM ones have already been learned: (1) it generates sensory goals (line 12) and queries what are the sensory delta movements that can achieve it (line 13), then (2) uses the previously learned SM maps to query what are the delta motor actions to observe those delta sensory movements (lines 14 and 15), and (3) updates only the sensory mapping (line 17).

2.7. Affordance gradients (AGs)

AGs were introduced in Sánchez-Fibla, Duff, and Verschure (2011) as object-centered SM structures to predict (bidirectionally) its sensory displacement effects given an actuation point on the object and a velocity vector. In Figure 2, we show the AGs learnt through systematic mobile robot pushes from different angles of a trapezoidal object as presented in Sánchez-Fibla et al. (2011). This raises the question of how are the egocentric and allocentric properties of the body–object interaction through affordances integrated. When seeing an object, we have access to its possible affordances not only independent of our body but independent of our mobility constraints. We compare the AGs learned in Sánchez-Fibla et al. (2011) with the ones here later on (shown in Figure 6). AGs are computed from sequences of SM interactions through the mutual covariation of groups of sensory points. Each sensory point has its area of reachability (as shown in Figure 3) and this area will determine and constrain the possible interactions of each sensory point.

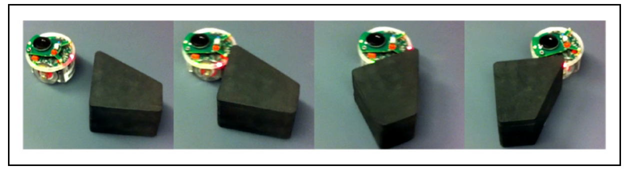

Figure 2. Epuck robot interacting with an object. In right column of Figure 6, we show the AGs learned for various objects (see Sánchez-Fibla et al., 2011, for further details).

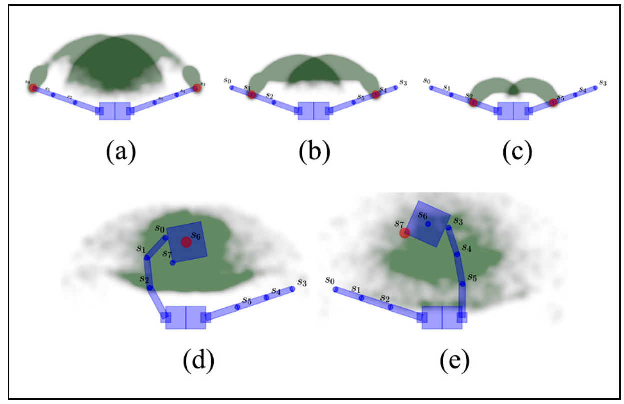

Figure 3. Reaching occupancy maps of the different SIPs after exploration. (a) s_0 and s_3 maps, (b) s_1 and s_4 maps, (c) s_2 and s_5 maps, (d) s_6 map, and (e) s_7 map. A Gaussian was added whenever an event caused an SM update.

AGs are defined here as sensory mappings and can be obtained from one point of actuation and a sensory displacement (mapping 3) or from two points (mapping 4, giving rise to bimanual affordances).

Consider, for example, mapping (3) where two SIP s_i,s_j interact (only one being an active point, s_i) and having the associated mapping: $f_{ij}^{\Delta s}(s_j, \Delta s_j, s_i) = \langle s_i, \Delta s_i \rangle$ which we define as the AG_{ij} of sensory point i given actuation point j.

An AG_{ijk} can also be defined from mapping (4) with three interacting sensory points (two of them being actuators): $f_{ijk}^{\Delta s}(s_j, s_k, \Delta s_j, \Delta s_k, s_i) = \langle s_i, \Delta s_i \rangle$.

3. Results

We present a diverse set of results that follow our approach of building a proof of concept of emerged bimanual affordances from SM interactions. We start in the order of increasing complexity benchmarking the different mappings from SM to solely sensory ones, deriving to body-independent forward/backward prediction functions that we have named AGs.

We do not present a full portfolio of results of the different types of SM learning that have been

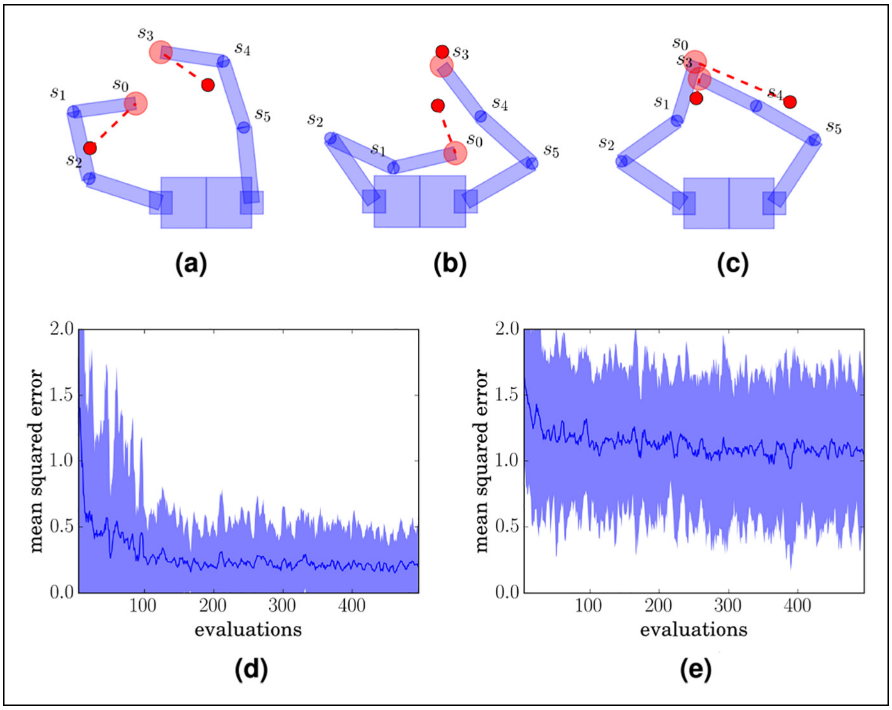

Figure 4. Learning of absolute and delta mappings. (a,b,c) Examples of bimanual reaching. The target and end-position of the absolute mapping are shown together with its associated error (dashed lines). (d) MSE learning curve of the absolute mapping $f^{abs}(m) = \langle s_0, s_3 \rangle$. (e) Mean squared error learning curve of the mapping: $f^{\Delta m}(m, \Delta m) = \langle m, s_0, s_3, \Delta s_0, \Delta s_3 \rangle$.

introduced: *Behavior-Based Babbling* and *Event-Based Learning*. We concentrate on the joint features that we tested and gave better results to achieve the learning of bimanual affordances. The main three steps toward bimanual manipulation that we benchmark are as follows: (1) the study of the delta mappings, (2) the AGs learned and applied to different SIPs, and (3) the bimanual AGs corresponding to learning the sensory mapping considering three SIPs.

3.1. Learning the delta SM mapping

We benchmark the absolute SM mapping in the case where we learn both arms at the same time $f^{abs}(m) = \langle s_0, s_3 \rangle$ and we compare it with the same motor to sensory delta mapping $f^{\Delta m}(m, \Delta m) = \langle m, s_0, s_3, \Delta s_0, \Delta s_3 \rangle$ (having the double DOFs both in the input and in the output spaces, indicating that it will be more difficult to learn).

For learning the absolute mapping, we perform 500 evaluations (see Figure 4(d)) of five goal positions during learning. Each goal is an end position of both arms (see Figure 4(a) to (c)). For every five tested goals, we plot the error mean and the minimum and maximum error values of the five evaluation tests (in a lighter color scale).

We use the *Context Goal Babbling* (Algorithm 1) for the delta case evaluated with an inverse MSE evaluation. For the learning, we perform 500 evaluations of five tested goals for a total of five delta movements generated at 500 absolute motor positions (see Figure 4(e)), thus a total of 2500 interest generated goals. For each five tested goals, we plot the minimum and maximum error values as before.

In many situations (positions of the arm m), achieving a delta movement is not possible. Many joint configurations make the arms collide when trying to be realized. See how the reachable space of both end-points intersect in Figure 3(a). It is also difficult to generate a uniform distribution of feasible delta movements for evaluation, and in fact delta goals are generated at random, being unfeasible many times.

3.2. Learning SIPs and object SM mappings

We present the mappings involving the object SIPs s_6 and s_7 learned from *Event-Based Learning* Algorithm 2 and *Behavior Based-Babbling* Algorithm 3. The number of tuples of each object model (learning is implemented with Explauto library using nearest neighbor, see section 2.3, executed in a quad-core i7 laptop computer during 3 h). The tuples stored for $f^{\Delta m}_{6,0}$ were 8553, for $f^{\Delta m}_{6,1}$ 1146, and for $f^{\Delta m}_{6,3}$ 567. As expected, it is higher for end-points and progressively decreasing for SIPs closer to the torso. This indicates the decreasing number of events related to s_0, s_1, and s_3. The number of tuples for s_0 seems quite high, but as we see in Figure 4, the error is quickly reduced in the beginning. The number

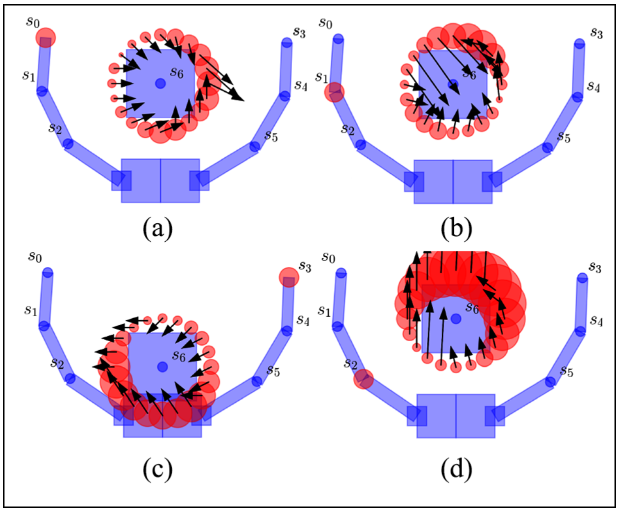

Figure 5. Affordance gradients $AG_{6,i}$ of the mappings $f^{\Delta s}_{6,i}(s_i, \Delta s_i, s_6) = \langle s_6, \Delta s_6 \rangle$ for $i \in \{0, 1, 2, 3\}$. (a, b) The object is at position (0, 1.5), (c) object position (0, 0.5), and (d) object position (0, 1.5).

of tuples of the nearest neighbor databases is not an exact number because we filtered the tuples that had no prediction error (line 9 of Algorithm 2).

We show results on the generated reaching maps (Figure 3). The reaching maps are generated adding a Gaussian, every time the SIP moves (as in Algorithm 2), to the map of the involved interest point, causing the movement (being in the causal context). These indicate where events happened.

3.2.1. AGs. We can compute the forward predictions of different mappings corresponding to the effect that different SIPs have on the object SIP s_6. Take, for example, the mapping of equation (3) specialized for s_6 and s_0: $f^{\Delta s}_{6,0}(s_0, \Delta s_0, s_6) = \langle s_6, \Delta s_6 \rangle$. We define the AG specialized for s_6 and s_0, $AG_{6,0}$ as the effects of delta changes Δs_0 applied toward the center of the object s_6 from all possible directions (shown in Figure 5(a)). From each incident position s_0, we also plot the certainty of that forward prediction by showing a bigger or smaller circle. The certainty corresponds to the normalized distance of the nearest point in the mapping. The uncertainty increases $AG_{6,2}$ (Figure 5(d)) in the upper right corner of the object. The prediction in an unobserved input sensory state is unreliable.

The computed AGs demonstrate that the object cannot be pulled toward the agent's body with one arm in certain conditions. Mapping $f^{\Delta s}_{6,0}$ suggests no option for bringing the object completely downward (Figure 5(a)) but still can be pulled towards. This impossibility gets accentuated with mapping $f^{\Delta s}_{6,1}$ (Figure 5(b)) and $f^{\Delta s}_{6,2}$ from which all movements push the object away. When learning mappings with both hands, the "pull toward" affordance becomes available in certain conditions.

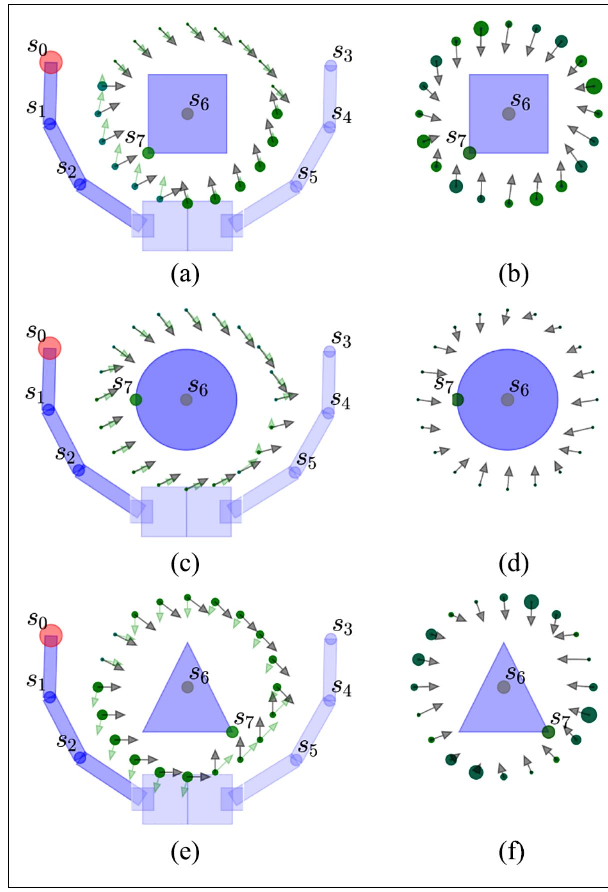

Figure 6. Learning AGs of other objects. Left column: AGs learned for s_0 when interacting with a (a) squared, (c) circular, and (e) triangular objects. At each point in the circle, the effect of applying a Δs_0 movement toward the center of the object is visualized as two vectors indicating the displacement of s_6 and s_7 (shaded) and a circle indicates how the object rotates (the bigger the circle the bigger the rotation). (c) No rotation is predicted for the circular object. Right column: AGs learned from a mobile robot interacting with the same objects (see Sánchez-Fibla et al., 2011, for details).

We compute the forward predictions of the mappings corresponding to the effect that different SIPs have on the two object SIPs s_6 and s_7. Take the mapping specialized for a particular case: $f^{\Delta s}_{6,7,0}(s_0, \Delta s_0, s_6, s_7) = \langle s_6, s_7, \Delta s_6, \Delta s_7 \rangle$. In this way, we can account for rotations of the object. After 500 interactions with the object, the mapping is queried to visualize the effects of Δs_0 actions moving toward the center of the object (see left column of Figure (6)). Figure 6(a) shows $AG_{6,0}$ for a squared object, Figure 6(b) shows $AG_{6,0}$ for a circular object, and Figure 6(b) shows $AG_{6,0}$ for a triangle.

We consider the AGs also in another case where a completely unconstrained approach/navigation can freely access any point of incidence of the object. It is the case of an Epuck robot interacting with objects by pushing actions (see Sánchez-Fibla et al., 2011, for

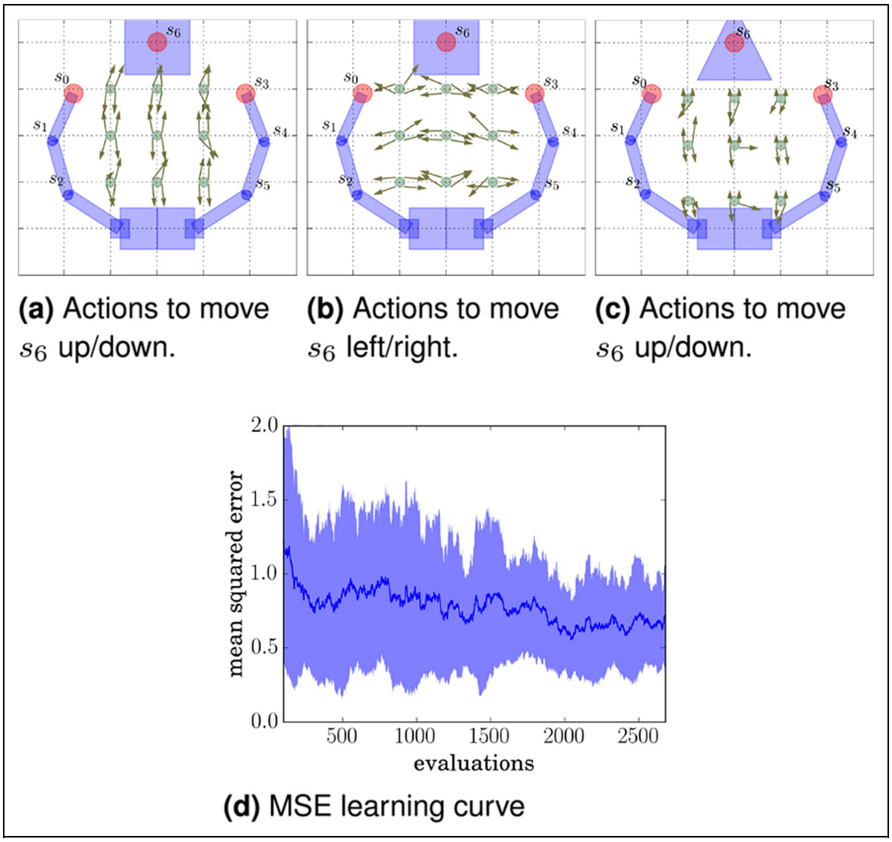

Figure 7. Learning of bimanual affordances. (a–c) Top row: Learned movements of the arm end-points to be performed for moving the object left/right (see figure b) and towards/away from the torso (see figures a,c). Bottom row: We plot the learning curve of the MSE error with the variance in a lighter color.

further details). For comparison purposes, we show side by side the AGs computed in the two cases (see Figure 6): freely moving mobile robot (right column) or a two-arm robot constrained by a fixed torso (left column, and setup of the paper). The comparison shows how AGs are body dependent as in the case of the mobile robot AGs are smooth and have symmetries.

3.2.2. Motor equivalence. In the sense of Bernstein, we describe the motor equivalence (available redundant motor programs to achieve the same sensory goal) that can be achieved by different mappings. As shown in Figure 5(a) and (b), a very similar effect (displacement of s_6) could be achieved by $AG_{6,0}$ (acting with s_0) or $AG_{6,1}$ (with s_1). Thus, the same sensory goal could be achieved with two alternative motor programs.

3.3. Bimanual AL

What are the mappings that we can use to learn bimanual affordances? The most general mapping that we can use is the three SIPs mapping that we have described. Depending on the effects that we want to capture on the object, we would need to consider two interest points s_6 (in the center) and also s_7 (in one corner) to be able to deal with the object rotations. This would lead us to learning the mapping: $f^{\Delta s}_{6,0,3}(s_0, s_3, \Delta s_0, \Delta s_3, s_6, s_7) = \langle s_6, s_7, \Delta s_6 \rangle$. Instead of addressing this general case, we limit the learning to a situation where the arms are reactively holding the object (as in Figure 8). This corresponds to the reactive guidance provided by the described *Behavior-Based Babbling*. In this way, the input space is reduced and we only consider the Δs_i of each arm end-point. We lose in this case the relative positions of the arm end-points and the object. The simplified mapping that we learn is $f^{\Delta s}_{6,0,3}(\Delta s_0, \Delta s_3, s_6) = \langle s_6, \Delta s_6, \alpha \rangle$.

We learn it with *Behavior-Based Learning* Algorithm 3 on the output space $\langle \Delta s_6, \alpha \rangle$ having s_6 (the position of the object) as context. We include the angle α of the object rotation instead of considering two interest points on the object. We learn a model of 5700 tuples but we also realized that only few 50 tuples are necessary to start to observe consistent bimanual affordances. This may be caused by the oversimplified mapping. We show the error learning curve in Figure 7(d). The error plotted corresponds to the mean squared difference of both the angle and the delta difference:

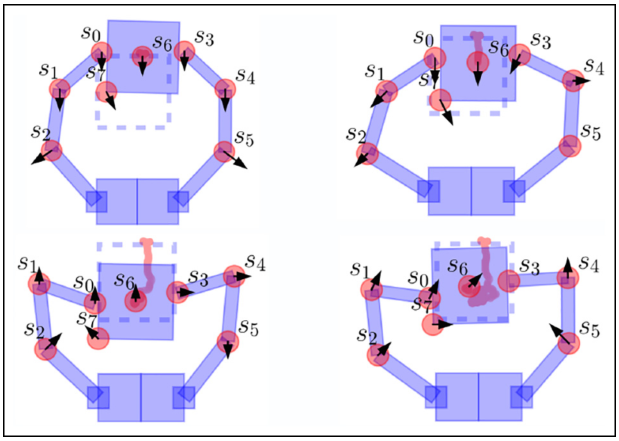

Figure 8. Bimanual control. Arbitrary sequence of actions in the peg-in-hole task.

$\langle \Delta s_6^{obs}, \alpha^{obs} \rangle - \langle \Delta s_6^{goal}, \alpha^{goal} \rangle$ being s_6 the provided context. The error is computed from delta Δs_6 goals that are randomized but fixing the target angle to 0.

To exploit the information of the SM mapping, or equivalently, of the AGs, to solve a peg-in-hole task, one has to query in an inverse way this mapping to obtain the delta actions of both end-points so that we attain a target displacement and target angle rotation (Δs_0 and Δs_3). To produce a target delta action, we can use the delta mapping learned for this purpose in an inverse way to obtain the target delta motor action: $g_0^{\Delta m}(m, s_0, \Delta s_0) = \Delta m$. Figure 7(a) and (b) show different inverse queries in different points of the sensory space to achieve the goals of moving the object up and down (first case) and moving it left and right (second case). The first case of moving the object up and down was previously impossible to achieve with two interest points mappings, as seen in Figure 5, in which no action gives the possibility of moving the object toward the robot from that position. Once the mappings are learned, we can set the goals in a greedy way so to match the target position of the object. Figure 8 shows a sequence of actions for a "pull toward the agent" target position.

We tried the learning with a triangular object and realized the reactive behavior needs to be changed to successfully be able to manipulate the object in any direction. In Figure 7(c), for example, we can see that the mappings learned do not perform any action when the object has to be pushed away, but correctly learned pull-toward actions when it has to be pushed down.

4. Discussion and related work

4.1. SM and solely sensory mappings

We raise an important issue for AL about the separation of SM mappings from the sensory ones, allowing the latter to be independent of changes in the confirmation of the body, more efficient in terms of being independent of the high number of motor DOFs (i.e. there are 700 skeletal muscles and tendons in the human body), and applicable when observing other agent's actions. In this way, the processes of fine tuning motor to sensory mappings can run in parallel while learning the covariations of related perceptual cues, bidirectionally boosting the acquisition of sensory to sensory ones which are fundamental to acquire basic SM contingencies based on physical laws (Battaglia, Hamrick, & Tenenbaum, 2013): gravity consequences, collisions, and so on. Here, we first learn the kinematics of the arms before learning any interaction with the object, but in Forestier and Oudeyer (2016b), different procedures are designed to learn both mappings in parallel. Once a relevant mapping between motor and sensory signals is learned, predictions can happen in the sensory domain (for achieving better generalization to unexpected events, see, for example, Maffei et al., 2017).

The separation of motor and sensory spaces lead us to two learning algorithms. *Context Goal Babbling* interleaves the generation of absolute and local sensory goals and learns SM mappings of type 2 (forward/inverse delta control). On the other hand, *Behavior-Based Babbling* reaches absolute states by reactive behavior guidance and generates delta sensory goals training pure sensory mappings of types 3 and 4 (singular and multiple interaction control). Both algorithms are based on goal babbling (GB), that is, learning is driven by generated goals that need to be achieved. GB is suited for learning in redundant spaces as many motor states can bring to the same sensory. It is more efficient to learn how to achieve a sensory state by learning one of its causal motor states, than trying many motor states that bring to the same sensory state (Benureau et al., 2014; Moulin-Frier and Oudeyer, 2013). We believe that this advantage is less important when mappings are in the sensory side as this redundancy is reduced (singular and multiple interaction control mappings).

As we said, we learn SM and sensory maps based on a supervised methodology of input–output samples and not guided by reward. The mappings could be stored and approximated with deep neural networks by using two different networks for the forward and inverse mappings.

4.2. Links to Psychology and neurobiology

There is evidence in developmental psychology that the brain goes from synchronous whole body control to asynchronous specialized control, from low DOFs to a higher DOF's control: as an example, Corbetta and Thelen (1996) show that synchronized bimanual movements are learned before unimanual ones during infant development. Bimanual movements may have large sensory consequences. This is a hint to the fact that the brain must encode the motor to sensory consequences and the learned covariations of perceptual cues separately. The newborn has to start making sense of the laws governing the visual modality as soon as possible.

Another consequence of this fact is that different specialized motor programs to realize the same goal-directed object interactions accumulate during development a characteristic that is known as motor equivalence (Sporns & Edelman, 1993), term coined by Bernstein. Motor program redundancy increases during development because an increasing set of specialized motor elements is learned to be used.

Neurobiological evidence shows that the brain encodes different schemes for unimanual and bimanual movements (Yokoi, Hirashima, & Nozaki, 2011, 2014), multiplicatively encoding information from the kinematics of both arms. This would be consistent with our hypothesis of the fact that we need a single and a multiple interaction mapping (sensory mappings 3 and 4) to perform bimanual movements. In the experiments section, we also benchmarked the SM map including a multiplicative encoding of all motors and a delta action to several output sensory-related points: $f^{\Delta m}(m, \Delta m) = \langle m, s_0, s_3, \Delta s_0, \Delta s_3 \rangle$. In this way, not only sensory maps are dedicated to the covariation of sensory perceptual cues, but SM maps as well.

4.3. Learning the mappings

Two families of exploration mechanisms recently shown to be efficient to learn complex non-linear redundant SM mappings. The first one concerns the space in which the learning agent chooses points to explore, what has been called the *choice space* in Moulin-Frier and Oudeyer (2013). Previous models (Baranes & Oudeyer, 2012; Rolf et al., 2010) have shown that learning redundant inverse models could be achieved more efficiently if exploration was driven by GB (choice space: $\mathcal{S}$), rather than direct motor babbling (choice space: $\mathcal{M}$). GB is especially efficient to learn highly redundant mappings (e.g. the inverse kinematics of a high-dimensional arm). At each time step, the agent chooses a goal s_g in the sensory space $\mathcal{S}$ (e.g. uniformly), uses the current knowledge of an inverse model to infer a motor command $m \in \mathcal{M}$ to reach s_g, observes the corresponding consequence $s = f(m)$, and updates its inverse model according to the newly collected (m, s) pair. This exploration strategy allows the agent to cover the goal space more efficiently, avoiding to waste time in redundant parts of the SM space (e.g. executing many motor commands that actually reach the same goal). The second mechanism comes from the field of active learning, where exploration strategies are conceived as an optimization process. Samples in the input space ($\mathcal{M}$ in our SM framework) are collected in order to minimize a given property of the learning process, for example, the uncertainty (Cohn, Ghahramani, & Jordan, 1996) or the prediction error (Thrun, 1995) of the model. This allows the agent to focus on parts of the SM space in which exploration is supposed to improve the quality of the model.

Combining both principles, recent works grounded in developmental psychology have concentrated on defining *empirical measures* of interest, either in the motor $\mathcal{M}$ or sensory $\mathcal{S}$ spaces. Computational studies have shown the importance of developmental mechanisms guiding exploration and learning in high-dimensional $\mathcal{M}$ and $\mathcal{S}$ spaces and with highly redundant and non-linear SM mappings (Baranes & Oudeyer, 2012; Oudeyer, Kaplan, & Hafner, 2007). Among these guiding mechanisms, intrinsic motivations, generating spontaneous exploration in humans (Berlyne, 1954; Deci & Ryan, 1985), have been transposed in curiosity-driven learning machines (Barto, Singh, & Chenatez, 2004; Schmidhuber, 1991; Schmidhuber, 2010) and robots (Baranes & Oudeyer, 2012; Oudeyer et al., 2007) and shown to yield highly efficient learning of inverse models in high-dimensional redundant SM spaces (Baranes & Oudeyer, 2010, 2012). Efficient versions of such mechanisms are based on the active choice of learning experiments that maximize learning *progress*, for example, improvement of predictions or of competences to reach goals (Oudeyer et al., 2007; Schmidhuber, 1991). This automatically drives the system to explore and learn first easy skills and then explore skills of progressively increasing complexity.

The *Behavior-Based Babbling* could also be guided by a process of synchronization and de-synchronization of DOFs. For instance, Kawai et al. (2013) presents a computational modeling of such a process: by synchronizing DOFs in control, we get quickly to good solutions that can then be fine-tuned by de-synchronization. In fact, some reflexes (like the grasp-reflex; Ugur, Nagai, Sahin, & Oztop, 2015) could be considered an early stage of synchronized control. So approaches that use multi-joint reflexes to guide learning could be affected by this fact (e.g. Sanchez-Fibla, Duff, & Verschure, 2013).

4.4. Delta mappings

Small changes in the motor or sensory signals can produce large effects in the sensory side (Bullock et al., 1993). A consequence of this fact is that learning of SM maps is best driven by the so-called delta actions: small changes in the motor or sensory signals. Computational models of affordances have considered the learning through delta actions (see, for example, Braud et al., 2018; Escobar-Juárez et al., 2016) closer to the continuous characterization of affordances, the so-called micro-affordances (Ellis & Tucker, 2000) in which a distinction between the brain triggered goal-related properties of objects (continuous action-effect possibilities; Sánchez-Fibla et al., 2011), and the motor programs to actually realize them (being initiated by hand (Hoff &

Arbib, 1993) or bimanual preshaping), are distinguished. Also, neurophysiological findings support the importance and usage of delta actions in the brain (see the gain field encoding of Yokoi et al., 2011).

We put emphasis on the fact that delta mappings can also be considered in the sensory side by capturing how sensory cues covariate. The so-called sensory delta mappings (types 3 and 4) capture the covariations of two and three SIPs, respectively. This fact is not contradicting Sensorimotor Contingency Theory (O'Regan & Noë, 2001). An action can also be characterized by its change in the sensory modalities, even forgetting the motor command that caused it, but it continues to be an action and something that actively moves/changes on the sensory side. One can then query an SM map to retrieve what is the motor action to be performed to observe a certain sensory change. The *Behavior-Based Babbling* Algorithm 3 generates sensory goals, queries sensory movements, acts using SM maps, learns only the sensory maps.

The learned sensory maps (types 3 and 4), and the derived so-called AGs, depend on the body, as the body facilitates different kinds of exploration. In Figure 5, we show how the affordances that an interaction enabled by end-point s_0 is different from the one of s_3. In Figure 6, we show that the affordances obtained from the constrained two-arm torso and the ones of a freely moving mobile robot are as well different.

4.5. RL

We previously discussed that the learning of the mappings is not guided by reward as RL methods are and that we are addressing the learning as a typical supervised problem. The mappings that we have presented, capture the consequences of delta (motor or sensory) actions and are indexed by goal information (goals to be achieved are queried as part of the forward or inverse input space of the mapping). This can be linked to the hierarchical RL approach of Kulkarni, Narasimhan, Saeedi, and Tenenbaum (2016) in which goals are also included in the RL function.

We separated the learning of SM and sensory maps. We performed a mixture of model-free (the SM maps), model-based (the sensory maps) learning. The sensory mappings serve as a model of the world that can guide the achievement of goals and the execution of actions. Model-based approaches are identified as the next direction to pursue in RL (Lake, Ullman, Tenenbaum, & Gershman, 2017).

Ideas related to GB have also links to RL. The fact that any sensory state achieved during learning can be used as goal state for the action that has just been performed is related to the concept of virtual goal of Baranes and Oudeyer (2012). This idea has been reinvented in the deep RL research: Hindsight Experience Reply (Andrychowicz et al., 2017).

4.6. Real-world dynamic environments

Our setup and results come with some limitations and could not be implemented in a real humanoid robot directly. First, we are assuming the agent has access to the sensory information through the sensory points (SIPs) that are given. SIPs should be computed from the visual image extracted from some stable features (as mentioned, one could use SIFT (Lowe, 1999) for that purpose). Alternatively, one could use some markers (QR codes) placed in the relevant point of the arms and objects. Second, we are also assuming a flat 2D environment implemented with a 2D simulation physics library. The extension to a three-dimensional (3D) world and 3D affordances is not clear and poses some challenges. From a sensory perspective, it can happen that the state is hidden or partially observable: an object may be moving because an actuation point behind it (and not visible) is applying a force to it. Zech et al. (2017) mention several papers dealing with a 3D environment extracting 3D features to compute affordances. If one wanted to implement our approach into a real robot directly without modification, we could assume a humanoid robot is looking downward to a table where the objects are placed and some markers are added on the objects and end-point effectors of the robot. The robot would need to know the plane of interaction with the table to make feasible movements. For the rest, we think that the setup captures some important aspects of bimanual interaction.

Another aspect that would complexify the learning is the presence of other agents. Other agents may cause events in the world that are not related to actions that we did: direct (arm moved) or indirectly (object moved). This fact is in favor of learning the SM and sensory maps separately. One would need to know what are the aspects of the world that we can control: see Kemp and Edsinger (2006) and Sánchez-Fibla, Moulin-Frier, Arsiwalla, and Verschure (2017) for approaches dealing with those aspects. Also learning of body image through haptic signals is important: see Mannella et al. (2018) where they use a very similar setup but only considering learning anticipatory haptic signals without the interaction with objects.

4.7. Tool use extensions

With a simpler simulated robotic arm, Forestier and Oudeyer (2016a) use a representation of SM experience as a hierarchy of SM models that a curiosity-driven agent explores. They study the role of this representation on the development of tool use precursors as the unfolding of three types of behaviors: behaviors without objects, behaviors with one object, and interaction between two objects.

5. Conclusion

We have looked at the low-level perspective of computationally modeling affordances and AL as a generalized case of forward/inverse kinematics learning. Our main contribution consists in separating the learning of SM mappings from the sensory covariations alone for multi-limb interaction. We have considered the following: (1) forward/inverse full body control and its body to absolute sensory mappings, (2) self-centered forward control (with the inclusion of delta sensory and motor actions), (3) singular interaction control (one to one sensory implications), and (4) multiple interaction control (how several sensory points affect each other, covariate). Mappings range from "motor to sensory" (types 1 and 2) to "sensory to sensory" (3 and 4). Sensory mappings do not need to be relearned if body changes. Mappings including knowledge of delta motor actions can be difficult to learn because they include the double of DOFs but at the same time provide a useful SM knowledge in the presence of a sensory/proprioceptive context. Deep neural networks could be used to approximate the mappings in an efficient way.

We make contributions to the learning mechanisms of the described SM and sensory maps. As small changes in the motor commands can produce large effects in the sensory domains, the mappings consider delta changes in the actions (for the motor spaces) and in the sensory domains (for the visual modality). The consideration of delta actions calls for a small generalization of GB to interleave absolute and local goals generation that we call *Context Based Babbling, Event-Based Learning* update of the different function-mappings that is activated whenever a sensory point moves and driven by active strategies (GB with progress monitoring). In the case of bimanual affordances, we added a reactive reflex guidance to attract the arms to the object, presented in *Behavior-Based Babbling* algorithm which performs a sensory-based learning and uses pre-learned SM maps to query the motor actions to be performed.

From the mappings we have extracted the so-called AGs, SM structures that can be queried in a forward or inverse way. We have observed that AGs are a good compression into usable forward and inverse prediction structures and can be smaller than the high number experiences stored in a SM mapping. We compute AGs for two cases: a completely unconstrained approach/navigation which can freely access any point of incidence of the object and another constrained body–object interaction in which arms and the fixed torso position limit the available affordances. For the former case, we have presented results with the mobile Epuck robot, which learns and exploits AGs (e.g. by an unconstrained version of the peg-in-hole problem presented in Sánchez-Fibla et al., 2011).

Solving the bimanual peg-in-hole task (in a custom simulation setup with self-collisions; see Note 1), we have faced the difficulty of the mutual interference of the different arms during learning. We reduced the problem to an SM learning task in which we departed from a reactive behavior where both arms maintain contact when haptically activated (we call this learning guidance *Behavior-Based Babbling*). This way, GB on the target direction and angle of the manipulated object, we have been able to solve the peg-in-hole task in this restricted set of cases. The "pull toward" affordance is discovered in this way. Clearly, the joint model of two arms has more affordances than the separate arm's models (a particularly placed distant object can only be approached toward the agent using both arms). But at the same time, two arms when linked by being (and forcing to be) in contact with a common object restrict their DOFs by the linked system that they form.

To conclude and from a developmental perspective, several questions have been raised that we may have started to answer. Does the brain need to maintain different absolute and delta mappings? Could the separate learning of the SM and sensory maps be the clue to how we make sense of the sensory covariations given that newborn start interacting with synchronized bimanual movements as Corbetta and Thelen (1996) suggest? Could newborn attraction to biological motion, as found by Simion et al. (2008), be a consequence of sensory co-variation maps at work? Could sensory maps be the main mechanism behind transfer learning from bi- to unimanual interactions? And what about motor equivalence? How much of the separated learned arm models is transferred to the joint arms model? Could a dominant arm effect emerge naturally from a computational learning process like the one we describe?

Authors' Note

Jordi-Ysard Puigbo is also affiliated with Pompeu Fabra University, Barcelona, Spain.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by INSOCO-DPI2016-80116-P and socSMC-641321H2020-FETPROACT-2014.

ORCID iD

Martí Sánchez-Fibla https://orcid.org/0000-0001-5725-1984

Note

1. The simulation environment (made with pyBox2D physics library), python notebooks to generate all results and

videos will be available at https://github.com/santmarti/2019-BimanualLearning-AB. The implementation depends on Explauto library https://github.com/flowersteam/explauto, for which a special learning with context has been implemented for this paper: see notebook learning_with_sensorimotor_context.ipynb.

References

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., & . . . Zaremba, W. (2017). Hindsight experience replay. In I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in neural information processing systems* (pp. 5048–5058). Long Beach, California, US: Neural Information Processing Systems Foundation, Inc.

Baranes, A., & Oudeyer, P. Y. (2010). *Intrinsically motivated goal exploration for active motor learning in robots: a case study*. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, Taipei, Taiwan. Retrieved from https://ieeexplore.ieee.org/document/5651385

Baranes, A., & Oudeyer, P. Y. (2012). Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, *61*, 49–73.

Barto, A., Singh, S., & Chenatez, N. (2004). Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings 3rd International Conference Dvp*. Learn (pp. 112–119). Cambridge, MA: MIT Press.

Battaglia, P. W., Hamrick, J. B., & Tenenbaum, J. B. (2013). Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, *110*, 18327–18332.

Benureau, F., Fudal, P., & Oudeyer, P. Y. (2014). Reusing motor commands to learn object interaction. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on IEEE*. Retrieved from https://hal.inria.fr/hal-01074822/document

Berlyne, D. E. (1954). A theory of human curiosity. *British Journal of Psychology*, *45*, 180–191.

Braud, R., Pitti, A., & Gaussier, P. (2018). A modular dynamic sensorimotor model for affordances learning, sequences planning, and tool-use. *IEEE Transactions on Cognitive and Developmental Systems*, *10*, 72–87.

Bullock, D., Grossberg, S., & Guenther, F. H. (1993). A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm. *Journal of Cognitive Neuroscience*, *5*, 408–435.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145.

Corbetta, D., & Thelen, E. (1996). The developmental origins of bimanual coordination: A dynamic perspective. *Journal of Experimental Psychology: Human Perception and Performance*, *22*(2), 502–522.

Deci, E., & Ryan, R. M. (1985). *Intrinsic Motivation and self-determination in human behavior*. New York, NY: Plenum Press.

Ellis, R., & Tucker, M. (2000). Micro-affordance: The potentiation of components of action by seen objects. *British Journal of Psychology*, *91*, 451–471.

Escobar-Juárez, E., Schillaci, G., Hermosillo-Valadez, J., & Lara-Guzmán, B. (2016). A self-organized internal models architecture for coding sensory–motor schemes. *Frontiers in Robotics and AI, 3*, Article 22.

Forestier, S., & Oudeyer, P. Y. (2016a). Curiosity-driven development of tool use precursors: A computational model. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *38th Annual Conference of the Cognitive Science Society* (CogSci 2016), Philadelphie, PA, United States, August 2016 (pp.1859–1864).

Forestier, S., & Oudeyer, P. Y. (2016b). Modular active curiosity-driven discovery of tool use. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on IEEE* (pp. 3965–3972). Daejeon, South Korea: IEEE.

Gibson, J. (1986). *The ecological approach to visual perception*. Mahwah, NJ: Lawrence Erlbaum.

Hoff, B., & Arbib, M. A. (1993). Models of trajectory formation and temporal interaction of reach and grasp. *Journal of Motor Behavior*, *25*, 175–192.

Hosoda, K., & Asada, M. (1994). Versatile visual servoing without knowledge of true jacobian. In *Intelligent Robots and Systems' 94. "Advanced Robotic Systems and the Real World," IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on IEEE* (*Vol. 1*, pp. 186–193). Munich, Germany: IEEE.

Kawai, Y., Park, J., Horii, T., Oshima, Y., Tanaka, K., Mori, H., & . . . Asada, M. (2013). Throwing skill optimization through synchronization and desynchronization of degree of freedom. In X. Chen, P. Stone, L. E. Sucar, & T. van der Zant (Eds.), *RoboCup 2012: Robot Soccer World Cup XVI* (pp. 178–189). Berlin, Germany: Springer.

Kemp, C. C., & Edsinger, A. (2006). What can I control?: The development of visual categories for a robots body and the world that it influences. In *IEEE International Conference on Development and Learning (ICDL), Special Session on Autonomous Mental Development*. Bloomington, IN, USA: IEEE.

Kulkarni, T. D., Narasimhan, K., Saeedi, A., & Tenenbaum, J. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems* (pp. 3675–3683). Barcelona, Spain: Neural Information Processing Systems Foundation, Inc.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, *40*, 1–72.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on IEEE* (*Vol. 2*, pp. 1150–1157). Kerkyra, Greece: IEEE.

Maffei, G., Herreros, I., Sanchez-Fibla, M., Friston, K. J., & Verschure, P. F. (2017). The perceptual shaping of anticipatory actions. *Proceedings of the Royal Society B*, *284*(1869). doi:10.1098/rspb.2017.1780

Mannella, F., Santucci, V. G., Somogyi, E., Jacquey, L., O'Regan, K., & Baldassarre, G. (2018). Know your body through intrinsic goals. *Frontiers in Neurorobotics 12*, Article 30.

Moulin-Frier, C., & Oudeyer, P. Y. (2013). Exploration strategies in developmental robotics: A unified probabilistic framework. In *Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on IEEE* (pp. 1–6). Osaka, Japan: IEEE.

Moulin-Frier, C., Rouanet, P., & Oudeyer, P. Y. (2014). *Explauto: An open-source Python library to study autonomous exploration in developmental robotics*. In *ICDL-Epirob-International Conference on Development and Learning*, Epirob. Genoa, Italy: IEEE.

Moulin-Frier, C., Sanchez-Fibla, M., & Verschure, P. F. (2015). *Autonomous development of turn-taking behaviors in agent populations: A computational study*. In *International Conference on Development and Learning, ICDL/Epirob*, Providence, RI. Providence, USA: IEEE.

O'Regan, J. K., & Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, *24*, 939–973.

Oudeyer, P. Y., Kaplan, F., & Hafner, V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, *11*, 265–286.

Puigbo, J. Y., Moulin-Frier, C., Vouloutsi, V., Sanchez-Fibla, M., Herreros, I., & Verschure, P. F. (2015). Skill refinement through cerebellar learning and human haptic feedback: An icub learning to paint experiment. In: *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on IEEE* (pp. 447–452). Seoul, South Korea: IEEE.

Rolf, M., & Asada, M. (2014). Autonomous development of goals: From generic rewards to goal and self detection. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on IEEE* (pp. 187–194). Genoa, Italy: IEEE.

Rolf, M., Steil, J. J., & Gienger, M. (2010). Goal babbling permits direct learning of inverse kinematics. *IEEE Transactions on Autonomous Mental Development*, *2*, 216–229.

Sánchez-Fibla, M., Duff, A., & Verschure, P. F. (2011). The acquisition of intentionally indexed and object centered affordance gradients: A biomimetic controller and mobile robotics benchmark. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on IEEE* (pp. 1115–1121), San Francisco.

Sanchez-Fibla, M., Duff, A., & Verschure, P. F. (2013). A sensorimotor account of visual and tactile integration for object categorization and grasping. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on IEEE* (pp. 107–112). Karlsruhe, Germany: IEEE.

Sánchez-Fibla, M., Moulin-Frier, C., Arsiwalla, X., & Verschure, P. (2017). A correlational analysis of multiagent sensorimotor interactions: Clustering autonomous and controllable entities. Retrieved from https://arxiv.org/abs/1711.08333

Schmidhuber, J. (1991). A possibility for implementing curiosity and boredom in model-building neural controllers. In J. A. Meyer, & S. W. Wilson (Eds.), *Proceedings of SAB'91* (pp. 222–227). Paris, France: MIT Press.

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, *2*, 230–247.

Simion, F., Regolin, L., & Bulf, H. (2008). A predisposition for biological motion in the newborn baby. *Proceedings of the National Academy of Sciences*, *105*, 809–813.

Sporns, O., & Edelman, G. M. (1993). Solving Bernstein's problem: A proposal for the development of coordinated movement by selection. *Child Development*, *64*, 960–981.

Takiyama, K., & Sakai, Y. (2016). Balanced motor primitive can explain generalization of motor learning effects between unimanual and bimanual movements. *Scientific Reports*, *6*, Article 23331.

Thill, S., Caligiore, D., Borghi, A. M., Ziemke, T., & Baldassarre, G. (2013). Theories and computational models of affordance and mirror systems: An integrative review. *Neuroscience & Biobehavioral Reviews*, *37*, 491–521.

Thrun, S. (1995). Exploration in active learning. In M. Arbib (Ed.), *The handbook of brain science and neural networks* (pp. 381–384). Retrieved from https://pdfs.semanticscholar.org/5014/f255f686913e21fd01cc53bf39d8db20ee1e.pdf

Ugur, E., Nagai, Y., Sahin, E., & Oztop, E. (2015). Staged development of robot skills: Behavior formation, affordance learning and imitation with motionese. *IEEE Transactions on Autonomous Mental Development*, *7*, 119–139. doi:10.1109/TAMD.2015.2426192

Verschure, P. F., Pennartz, C. M., & Pezzulo, G. (2014). The why, what, where, when and how of goal-directed choice: Neuronal and computational principles. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *369*(1655), 20130483.

Verschure, P. F., Voegtlin, T., & Douglas, R. J. (2003). Environmentally mediated synergy between perception and behaviour in mobile robots. *Nature*, *425*, 620–624.

Yokoi, A., Hirashima, M., & Nozaki, D. (2011). Gain field encoding of the kinematics of both arms in the internal model enables flexible bimanual action. *Journal of Neuroscience*, *31*, 17058–17068.

Yokoi, A., Hirashima, M., & Nozaki, D. (2014). Lateralized sensitivity of motor memories to the kinematics of the opposite arm reveals functional specialization during bimanual actions. *Journal of Neuroscience*, *34*, 9141–9151.

Zech, P., Haller, S., Lakani, S. R., Ridge, B., Ugur, E., & Piater, J. (2017). Computational models of affordance in robotics: A taxonomy and systematic classification. *Adaptive Behavior*, *25*, 235–271.

About the Authors

Martí Sánchez-Fibla is currently leading Plan Nacional INSOCO DPI2016-80116-P and working on studying emergent behaviors among learning agents. Martí is professor and coordinator of UPF master Cognitive Sciences and Interactive Media (CSIM) affiliated to the Artificial Intelligence and Machine Learning group of University Pompeu Fabra (UPF), collaborating with SPECS (IBEC) and Complex System (UPF) Labs. Martí is a computer scientist and did his PhD thesis in combinatorial optimization in the Artificial Intelligence Institute (IIIA, CSIC).

Sébastien Forestier is a PhD student in Developmental Robotics at Inria Bordeaux under the supervision of Pierre-Yves Oudeyer. He studies robotic models of the development of speech and tool use in human infants to better understand the role of intrinsic motivations in the development of those skills and to design efficient curiosity-driven learning algorithms for robots.

Clément Moulin-Frier's main research interest is the modeling of communication emergence in sensorimotor agents. He is currently a Research Scientist in the AI company Cogitai, Inc. After a PhD in Engineering of Cognition, Interaction, Learning and Creation obtained in 2011 at Grenoble University (France), he worked in different research groups across Europe and the United States. These include the Flowers group at Inria (France), the SPECS group at the University Pompeu Fabra (Spain), the University of Southern California (USA), and the Collège de France (France). He has published approximately 30 papers in international journals and conferences. In 2015, he was Program Chair of the ICDL-Epirob conference in Brown University, USA. He is the second coordinator of the INSOCO project with Martí Sánchez-Fibla.

Jordi-Ysard Puigbò, trained as an engineer, is a researcher at the intersection between neuroscience and machine learning. Mostly interested in learning rules, mechanisms, and architectures, his research is centered in understanding the differences between learning from reward and punishment, how and why the latter is usually faster in biological systems, and how machine learning can benefit from the insights that neuroscience has matured during the last 20 years.

Paul FMJ Verschure leads the Synthetic Perceptive, Emotive and Cognitive Systems Laboratory, hosted by the Institute for Bioengineering of Catalunya. Paul has received his MA and PhD in Psychology and pursued his research at different leading institutes in Europe and the United States and manages a team of over 30 researchers and technicians with whom he has published over 350 articles in leading journals and conferences in a range of disciplines. Paul is regularly invited as a speaker at relevant scientific conferences and international policy events, a consultant for the European Commission, board member of three journals, and reviews for a number of relevant journals and conferences. Paul is founder/CEO of Eodyne Systems S.L, which is commercializing neurorehabilitation, education, and cultural heritage technologies, founder/Chairman of the Future Memory Foundation, and founder/Chairman of the Convergent Science Network Foundation.