A modular Dynamic Sensorimotor Model for affordances learning, sequences planning and tool-use

Raphaël Braud, Alexandre Pitti, Philippe Gaussier

A modular Dynamic Sensorimotor Model for affordances learning, sequences planning and tool-use

Raphael Braud*, Alexandre Pitti and Philippe Gaussier

***Abstract*—This paper proposes a computational model for learning robot control and sequence planning based on the ideomotor principle. This model encodes covariation laws between sensors and motors in a modular fashion and exploits these primitive skills to build complex action sequences, potentially involving tool-use. Implemented for a robotic arm, the model starts with raw unlabelled sensor and motor vectors and autonomously assigns functions to neutral objects in the environment. Our experimental evaluation highlights the emergent properties of such a modular system and we discuss their consequences from ideomotor and sensorimotor-theoretic perspectives.**

***Index Terms*—Sensorimotor Laws, Affordance, Sequences, Tool-Use, Ideomotor principle, Robotics**

I. Introduction

Inspired from the study of infant development, conceptual models of sensorimotor control have highlighted that behaviours are mostly goal-directed. In this view, the ideomotor principle [1] postulates that action and action planning are controlled by an anticipatory representation of the desired effect. Common coding specifically proposes that actions are coded in terms of their perceptual effects [2]. Considering the fact that infants clearly grow through open-ended acquisition of novel behaviours, the ideomotor principle suggests a coupling with computational models that:

- Involve cost (or reward) functions that are not dedicated to a particular task [3], [4]
- Favour reusable skills that are directly applicable to unseen scenarios [5]
- Acquire new skills while retaining those that were acquired through past experiences [6] (*i.e.*, incremental learning).

In this paper, we propose such a model, inspired by theoretical models developed by Pezzulo and Castelfranchi [7].

Goal-directed behaviours raise the question of going beyond an immediate reaction to the environment, that is a mere mapping of perceptions to actions. Pezzulo and Castelfranchi formulate theoretical models based on the prediction of action consequences, insisting on the close relationship between mental imagery and the motor system. They aim to go beyond the learning of forward and inverse models as distinct mechanisms and suggest a bidirectional learning of actions and effects.

Laboratory ETIS, UMR CNRS 8051, University of Cergy-Pontoise, ENSEA, France

* Corresponding author: raphael.braud@gmail.com

The properties discussed above have to be combined with a developmental perspective. First, it appears that infant development entails systematic and radical changes in multimodal experiences and sensory correlations. This favours the idea that development requires to continuously extend existing skills to new inputs [8]. A stronger argument states that sensory input is initially unlabelled. That is, the specific role of each modality is unknown *a priori*. The interplay between sensory input must then be discovered through an enactive process. This major concern has been well illustrated by Denett [9]. He invites us to imagine being imprisoned in a windowless control center of a giant robot, where the surrounding walls are covered with sensors (i.e., lamps) and effectors, none of which are labelled. The final question is not only to create meaningful labels, but also to control the robot's behavioural repertoire. More practically, Pierce and Kuipers consider a learning agent equipped with an uninterpreted sensorimotor apparatus [10]. The agent has no *a priori* knowledge about the sensory and motor system, the sensors' meaning or the effect of motor signals. The apparatus is considered as a raw sensory vector and a raw motor vector. The authors claim that raw values are not directly suitable for describing the structure of the world and for predicting the effect of actions on sensors, thus requiring an abstraction stage. The latter is obtained by generating features (e.g., using PCA), assuming approximately linear relationships between action magnitudes and feature derivatives [10]. A similar hypothesis is explored in the ASAMI model [11] or in information-theoretic frameworks [12], [13].

Second, it remains unclear to what extent low-level skills (e.g., sensorimotor mappings) and higher-level cognitive abilities (e.g., action planning, tool-use) share common learning mechanisms. The internal dynamics of the agent and the dynamics of the environment, coupled during the acquisition of sensorimotor skills, may allow both low- and high-level skills to share a common metric space [14]. This advocates that a continuous development between different levels of cognitive abilities can lead to complex behaviours such as the use of tools [15], [16]. In a developmental framework, low-level behaviours can clearly form atomic components of high-level learning [11], [17].

This paper proposes a computational model that starts with raw sensors and motors and provides; (i) an incremental learning of the derivative effect motors have on sensors (forward control model) and (ii) an open-ended mechanism that uses the forward model to solve unseen tasks on-the-fly and on

demand. The proposed model acquires low-level and higher-level skills through identical learning mechanisms.

This work provides a novel instantiation of concepts developed by Pezzulo and Castelfranchi. The proposed model predicts sensor variations based on sensor-based contexts and motor magnitudes, where all values are continuous. The sensors are not categorised according to their sensor space and its topology. Rather they categorised through actions in order to learn sensorimotor laws. We propose a common coding between motor and sensor variations, such that a hierarchical encoding and planning is achieved, allowing the generation of action sequences.

The paper is organised as follows. Related works and principles are presented in section II. In section III, we describe a novel control architecture called Dynamic Sensori-Motor model. In section IV we report on our experimental results, whereas in section V we discuss related approaches before concluding the paper.

II. Related work

A. Tool-Use: a key example

Tool-use is a good illustration of the issues we have identified, as it addresses a wide range of problems.

Regarding low-level issues, tool-use involves body schema extension[1] as well as reaching or grasping tasks. It also deals with much higher level cognitive skills and abstraction capabilities [20], such as the ability to make sequences [21], [22], [23]. In addition, it is strongly related to language issues [24], [25], [26], [27].

In Fagard *et al* experiments, children have to use a rake-like tool to touch toys out of their reach. The results indicate that it is only between 16 and 20 months when infants start to intentionally try to bring the toy closer with a tool, suggesting that a true understanding of the use of a tool is not fully acquired before that age [28]. Among the striking capabilities an infant develops during its early infancy, we notice the means-end behaviour that appears approximately between 8 to 12 months and involves the deliberate and planned execution of a sequence of steps in order to achieve a goal. The behaviour occurs in situations where an obstacle must be first removed in order to achieve the goal [29]. This behaviour constitutes a very important step towards the use of tools, rendering it a requirement. Despite the high-level cognitive capabilities that are necessary for the means-end behaviour to occur, they are not yet sufficient to accommodate tool-use.

Experimental results from brain imaging on subjects that learn to make stone tools show that abstract conceptualisation or strategic action planning are not the central factors for developing tool-use, rather it implies that the sensorimotor adaptation and affordances perception play an important role [30]. In that context, tool-use can be deemed as a continuous developmental achievement where the main issue is to detect and relate affordances (please see [16] and [31] for an implementation on a real robot).

[1]Please see [18] and [19] for a review on how a body schema can be encoded in a robot.

Thus, this work focuses on the ability to encode sensorimotor skills in a way that an open-ended use of affordances and sequences of actions become feasible.

B. Sensorimotor encoding in the Robotic field

In robotics, the effectiveness of tackling the problem of encoding sensorimotor information is well illustrated by encoding both motor and end-effector coordinates (e.g., for the completion of a reaching task). Roboticists are faced with the motor equivalence problem, which states that a redundant arm has an infinite number of ways to reach towards a given target. That is, a defined target can be reached using multiple motor trajectories. The latter holds for any effector system with a higher dimensionality than the target specification. Two main strategies for sensorimotor encoding of both motor and end-effector coordinates are considered for accomplishing reaching tasks [32], [33].

The first one relies on learning mappings between motor and end-effector coordinates. The effector system can then reach a given target X_B by activating the motor coordinates θ_B associated with the desired end-effector coordinates.

$$\theta_B = f(X_B) \tag{1}$$

We call this strategy the absolute strategy, because absolute mappings are learned. An absolute strategy algorithm can for example be used as a homoeostatic system: after learning the associations, regardless of noise, the system is forced to stabilise its output to an equilibrium point based on the learned associations (or categories). Stabilisation occurs every time there is a conflict within its associated inputs. These categories can then become visio-motor attractors, for example, with both vision and proprioceptive inputs, as seen in [34], [35]. Here, it is worth noticing that in absolute strategies an algorithm dedicated to the minimisation of the distance between the current and the desired position is required (e.g., a PID). If a robot has to perform a continuous sequence of end-effector coordinates (forming an "8" for example), it can generate a continuous sequence of motor configurations.

The second strategy depends on the learning of mappings between each spatial direction of the end-effector and the changes in the joint angles that cause the movement in the spatial direction. Typical approaches involve a direct model, through the Jacobian matrix J:

$$\dot{X} = J(\theta)\dot{\theta} \tag{2}$$

We call this a relative strategy. Unlike the absolute one, the relative strategy allows an effector system to reach a target by utilising the direction from the end-effector X_A to the target X_B. This direction is associated with the learned variation required to be applied to the joint angles.

$$\dot{\theta}_B = f(X_B - X_A) \tag{3}$$

Such function is usually estimated by the use of dedicated inverse learning algorithms. Typical methods compute a generalised inverse of the Jacobian matrix [36], [37] and [38] to

ultimately select one particular solution among all those that are available, in order to perform a defined task.

$$\dot{\theta} = J^{+}(\theta)\dot{X} \tag{4}$$

An advantage of the relative strategy is seen in the way it addresses the geometry transformations of a robot such as the extension of the arm that occurs in tool-use [32], [33]. Most importantly for this work, the relative strategy does not directly learn the target sensor configuration, rather it learns the whole diversity of means available to the system in order to achieve a reaching task. This is highly beneficial, as the system is capable of reusing what it has previously learned with any action selection mechanism as a repertoire of valid means for different ends. This strategy fosters a clear separation between *what* is learned and *how* learning can be used.

Based on the idea that the world is modular, Wolpert and Kawato proposed a model based on paired forward and inverse models [39]. According to their proposition, forward models learn the causal relationship between actions and their consequences, and in turn can be used as predictors or simulators of the consequences of actions [40].

A modular representation of the world is efficient when there is a clear independence between modules. This typically holds in linear systems. However, as most components of real-world systems exhibit non-linear and dynamic interactions, defining a specific modularity pattern affects the dynamics of the whole system. Although modules are defined according to task-related factors (e.g., the interactions with a particular object or within a particular environment), they can be employed for closely related tasks. However, modules may become less effective when the factors that define them are not met in a particular task and exhibit different non-linear dynamic interactions. In this case the utilisation of previously defined modules may render the system less accurate and, in turn, may become a problem during the modulation of the contribution of the inverse models.

It is argued that predictive learning of sensorimotor information plays a key role in cognitive development, and may be able to produce a continuous development between different levels of cognitive abilities [15]. Rolf and Asada also point out the interesting role of goals in developmental robotics and autonomous mental development [41]. Goals are often handcrafted and chosen by a designer: they materialise the central question for open-ended development, particularly when goals cannot be known in advance. It raises questions about how they could be learned autonomously, through perceptions. For Rolf and Asada, goals can often be easily interpreted in terms of underlying affordances. They describe discrepancies between a present state and an imagined and desired state, while affordances describe general possibilities that can be positively valued by goals. This mechanism is described in the next section.

C. Sensorimotor encoding, goals and affordances in psychology and neuroscience fields

As regards the ideomotor principle [1], goals are seen as anticipatory representations of the expected action effects, and voluntary actions are represented and controlled by their anticipated action effects. This idea of encoding actions in terms of goals is further developed in the common coding theory [2], where perceived events and actions generated by distal events are coded and stored together in one common representational domain. In the Theory of Event Coding [42], perceiving a stimulating object and planning a voluntary action are not distinct processes operating on completely different codes. Rather, they are functionally equivalent, because both are dealing with an internal representation of external events. Hommel also defends that such coding is somehow hierarchical and relies on events that could be segmented into several meaningful units.

In fact, planning an action raises the question of how sequences of meaningful units of actions are organised. Seminal works [43], [44] argue for a hierarchical structure in human action. A computational model of a hierarchically organised network of action schemas, mapping onto the structure of the relevant task domain (e.g., task of coffee-making) was proposed by Cooper and Shallice [45]. A review is found in [46]. In neuroscience, hierarchical architectures that drive their inspiration from the prefrontal cortex (but relying on multiple neural circuits) are proposed for planning goal-directed sequences [47], [48]. Similarly in the robotics field, there exist several works that propose models of a hierarchical structure of behaviour [49], [50], [51], [52]. For Bonini *et al.*, ventral premotor and inferior parietal neurons together with the prefrontal cortex can code the goal of intentional actions at different levels of motor abstraction. In addition, parieto-premotor circuits along with the prefrontal cortex are responsible for the organisation of motor acts into action sequences and for keeping the internal representations of the individual motor intention active [53].

Moreover, the discovery of mirror neurons on macaque monkeys in brain area F5 (considered as homologous to human posterior inferior frontal cortex) point to a mechanism of representation of an action goal (see [54], [55] for a review). In fact, mirror neurons not only respond to specific actions, e.g., the grasping of objects, but also they respond to the observation of another monkey or a human performing the same goal-directed action. More precisely, mirror neurons respond differently when the same observed act is embedded in a specific action, with a different motor chain (grasping for eating, grasping for placing, etc.) and a different intention, suggesting that they are sensitive to goals at different levels (proximal or distal goals) [56]. Canonical neurons are also discovered in the same brain area. In addition to responding to specific actions like grasping an object, they also respond to the mere observation of an object [54]. This suggests a mechanism of affordances. The concept of affordances was first introduced by Gibson [57], as resources that the environment offers to any animal that has the ability to perceive and make use of them. From this ecologist perspective the perception of affordances is direct, unmediated by neural or mental representations. From the representationalist perspective however, affordances are internalised through mental representations for use by computational processes, for example with a sensorimotor agent interacting with the world. Tools, for instance, are

manipulable objects that elicit multiple affordances, in relation to their shape or their function ([58], [59]). Interestingly, Ellis and Tucker proposed the concept of micro-affordances, for possible interactions not with the whole object but with rather specific action components. For example a given size, shape or orientation of an object will trigger partial activations of the motor patterns required to interact with it [60]. Since the environment continuously provides the brain with a great number of opportunities for actions, it highlights the problem of interference between them, and more specifically the problem of selecting among them. In the affordance competition hypothesis developed by Cisek [61], [62], the brain processes sensory information and specifies several potential actions that are currently available, i.e., the affordances, in parallel. Another major pathway in the brain is supposed to bias the selection among the available affordances, which compete against each other. In a recent work, Thill *et al.* review such mechanisms and their related computational models, and put emphasis on the role of goals and more specifically the importance of the prefrontal cortex in this selection [63].

Considering the idea that affordances depend on actions from the observer, Moller *et al.* assume that affordances are revealed by a process of internal simulation of action consequences [64]. In order to perform such simulation, the ability to anticipate is necessary. Anticipation is a basic computational mechanism in the brain and plays an significant role [65], [66]. The anticipation of the effects of an action is usually done through internal, both forward and inverse, models. A forward model requires the integration of perceptual states and efferent copies of motor commands in order to predict the sensory consequences of the actions, whereas the inverse model requires the actual and desired states in order to generate motor commands to achieve the desired state (see [67], [68], [69], [70]). Morover, models with simulated inputs can be used for internal simulations. In the simulation theory, it is argued that thanks to such mental simulations, the same neural structures can underlie sensorimotor, cognitive and social abilities (such as self-detection, self-other distinction, planning, perception, mindreading, and imitation) [71], [72], [73], [74], [75]. For Moller, affordances are simulations of actions through a learned forward model. It is possible to predict and choose actions based on such simulations. Nevertheless, the multitude of sensorimotor sequences is too rich to be explored, so an inverse model is learned in order to reduce the exploration and to allow the immediate proposition of actions. Due to an evaluation of predicted situations, actions are labelled as "good" or "bad", and subsequently the inverse model is able to provide a restricted subset of promising sequences [64].

The simulation theory not only provides an explanatory mechanism for affordances [64], but it also consists of a way to interpret the bidirectional links between goal representations and motor actions, as outlined by the ideomotor principle, extended through the common coding theory. For Pezzulo and Castelfranchi, *thinking consists in the control of imagination*, as the capability to control mental simulations to set up interactive subgoals and plans that achieve intentions beyond the here-and-now of perception [7]. Similarly to [64], they propose a parallel extraction of many affordances available in the environment by running internal simulations of possible actions. As in [61], they also propose a mechanism for selecting one of the available affordances based on the their values and achievability. They finally suggest two inhibition mechanisms; i) one that inhibits responses that are automatically triggered by affordances, and ii) one that allows the temporary suppression of both control commands and external inputs, and instead utilise internally generated (simulated) stimuli [7]. They also point to a working memory mechanism responsible for storing distal goals and subgoals, over time.

In line with the described ideomotor principles and with the sensorimotor theory [76], we combined an action based perception mechanism with a common coding between motors and sensors, by encoding sensor variations as motor magnitudes through locally linear associations of the two. In the Dynamic Sensorimotor Model (DSM), presented in the next section, a "motor" is defined as having two distinct characteristics. First, a motor is a magnitude and thus is able to drive changes in the sensor space. Second, this magnitude can possibly be manipulated by the robot. The first motors to be considered are the actuators of the robot, and their magnitudes are given by motor commands, e.g., in velocity. However, a sensor variation magnitude is tautologically making a change in the sensor space. Hence, once a control model is able to cause such a variation, it is the sensor variation that is considered as a motor in the context of DSM. Perceiving a motor as a sensor variation allows the robot to possibly consider any predictable sensor variations as motors, internal (e.g., proprioception) or external (e.g., the visual perception of a tool).

III. The Dynamic Sensorimotor Model

In order to be able to perform high-level tasks in an open-ended manner, it is argued that a robot should have the ability to recombine simple skills in order to perform a new task. Based on this, we propose a model where only basic skills are learned. In DSM, a basic skill describes the ability to control only one sensor. We then provide a mechanism to allow different skills to be combined on-the-fly, with respect to the constraints of both the task and the context.

In the next sections, III-A and III-B, we describe how a basic skill is built. In section III-C we show how they can be combined to achieve a task.

A. Encoding basic skills: the Sensorimotor Law Encoder

Following the ideomotor principle, the aim is to make sensory variations perceived as equivalent to motor magnitudes. We propose a relative strategy (described in section II-B) by which our Sensorimotor Law Encoder (SLE) learns how to predict sensory variations dS based on motor magnitude M, assuming a locally linear relationship between them. A comparable idea is exploited in previous research, where joint distribution of sensorimotor variables, known as gain-field neurons, are used to encode sensory variation based on prediction errors [77]. Considering that our model only tries to reduce dS to M, it is assumed that all observed changes in the

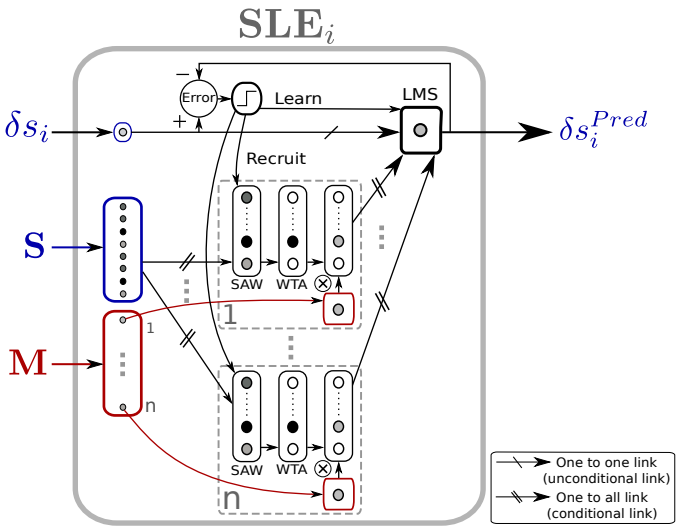


Fig. 1. Sensorimotor Law Encoder of sensor i. A learning predictor of the variation of sensor i that takes as input the vectors $\mathbf{S}$ and $\mathbf{M}$ of all sensors and motors, respectively. A categorisation of $\mathbf{S}$ is done to encode the contexts for each motor. The recruitment of a new context is based on the prediction error. The winning context is multiplied by one motor in a neuron. This neuron is then associated with the prediction of the variation of the sensor i.

environment are possibly a result of the robot's actions within it (see [10], [12] for examples of more developed works that consider this issue). In fact, DSM is based on the idea that there is no *a priori* separation between the robot's body and the world, which is shown to be beneficial when interpreting a tool as an extension of its body.

Considering the above, the input vector $\mathbf{S}$ is populated with sensors that concern both internal (e.g., proprioception) and external (e.g., vision) sensory data.

DSM's predictors, SLEs, build on the pioneering works of Bullock et al. [32], [78], [79]. There are three fundamental differences, which are designed to overcome difficulties found in the previously proposed models. First, DSM allows the learning of as many SLEs as sensors in the system, in order to be able to ultimately act on their corresponding sensors individually and change them independently, a technique which promotes the generation of different combinations of sensor variations. Next, each SLE should, in theory, receive all sensors as input. Although not all sensors are relevant to the prediction of just one sensor variation, in early stages of autonomous development, unexpected inputs can have the potential of being key predictors in the future. For instance, in our experiment tactile information is used to predict a variation of the hand position (with a tool). The importance of the tactile sensory information is not obvious when predicting the hand position, but during development *a priori* irrelevant input may be used to learn sensorimotor laws, due to unpredictable circumstances. In a similar vain, we also except that at some point during the development, a wrong, imprecise or incomplete law can be learned (e.g., tactile information is probably not enough, whereas vision should probably be taken into account when recognising a tool). This mechanism, although prone to generating laws that consider irrelevant sensors at early stages, it conceptually aligned with the effect developmental learning is expected to be [80]. Finally, the last difference is that our SLEs do not learn an inverse model, rather a simulation mechanism is used. The latter is discussed in detail in later sections.

In more detail, each SLE is dedicated to one sensor and is responsible for predicting its variation, i.e., δs_i for the sensor i. $\mathbf{M}$ is a copy of the vector of the N motor magnitude that is being currently executed: each motor m_n can potentially predict a variation of s_i depending on the context given by $\mathbf{S}$, the vector of input sensors. As depicted in fig. 1, the N motors are processed as follows. To start with, $\mathbf{S}$ is categorised using a Selective Adaptative Winner (SAW) neural network. The activation rule of the SAW is given by:

$$A_k = 1 - \frac{1}{N}\sum_j |w_{kj} - e_j| \quad (5)$$

where A_k is the activity of the k^{th} neuron of the output layer of the SAW, w_{kj} the weight of the connection between an output neuron k and the j^{th} neuron of the input layer whose activity is e_j. The connection's weight is set to e_j when the recruitment occurs (line 7 in **Algorithm 1**). There is no adaptation in our experiments, however an adaptation rule based on the K-Means algorithm may be added to adapt the weights of the winner. Considering eq.5, the closest the current inputs are to the learned inputs, the highest the activity A_k. Handling a recruitment is depicted in **algorithm 1** lines 6 to 8. Note that for a recruitment to occur two conditions need to be met. An error is detected on sensor i and one unique motor na has been activated since the time previous sensor values were stored (in s^{Stored}). A new category is then recruited within the corresponding SAW (sensor i, motor na).

The second step of processing of the motors consists of the selection of the most appropriate sensory category, using a Winner Takes All (WTA) neural network.

Finally, once selected, the winning neuron is multiplied by the scalar motor activity m_n. Each of those N multiplied neurons is then used as a conditional signal to learn the linear relationship between the motor m_n and the unconditional δs_i input. This is achieved by utilising a least mean square (LMS) algorithm. The contributions of each motor activity are then summed in the LMS to compute the predicted variation δs_i^{Pred}.

Notice that the learning signal, coming from the prediction error, is used both for the recruitment of the SAW and for the learning of the LMS. In addition, the triggering of the learning signal does not depend on the input sensor space but on a threshold Th_1 applied to the prediction error of each SLE (see fig. 1). Moreover, to avoid noise on instant derivative measures of the sensors, the error is not measured at every time iteration. Rather, it is only measured when enough change to the predicted sensor has been noticed, given an arbitrary threshold Th_1^{Step}. If the latter is too low, SLEs may be too sensitive to noisy input sensors and may recruit too many neurons. If the threshold is too high, the learning may not be accurate enough. The derivative sensor value to learn Δs_i^{Real}, in input of each SLE, is then integrated on a meaningful period.

Algorithm 1 : $\mathbf{SLE}_i$ (i, $\mathbf{S}^{Real}$, $\mathbf{M}^{Real}$)

$s_i^{Stored} \leftarrow s_i^{Real}$; $\Delta s_i^{Pred} \leftarrow 0$; $\forall n,\ M_n \leftarrow 0$
loop
  $\Delta s_i^{Real} \leftarrow s_i^{Real} - s_i^{Stored}$
  if $|\Delta s_i^{Real}| > Th_1^{Step}$ **or** $|\Delta s_i^{Pred}| > Th_1^{Step}$ **then**
    if $|\Delta s_i^{Pred} - \Delta s_i^{Real}| > Th_1^{Err}$ **then**
      // Error: Recruitment for the (unique) activated motor na, *i.e.* where $M_n \neq 0$, of the k^{th} neuron in SAW^{na}, and learning in LMS^{na}
      $\forall j,\ w_{kj}^{SAW^{na}} \leftarrow s_j^{Real}$
      $w_k^{LMS^{na}} \leftarrow \frac{\Delta s_i^{Real}}{M_{na}}$
    end if
    // New step: re-initialisation of the predictor
    $s_i^{Stored} \leftarrow s_i^{Real}$; $\Delta s_i^{Pred} \leftarrow 0$; $\forall n,\ M_n \leftarrow 0$
  end if
  // Perform current prediction
  $\forall n,\ M_n \leftarrow M_n + m_n$
  $\forall n,\ \begin{cases} \forall k,\ A_k^n \leftarrow 1 - \frac{1}{N}\sum_j |w_{kj}^{SAW^n} - e_j| \\ \forall k,\ B_k^n \leftarrow m_n A_k^n \end{cases}$
  $\delta s_i^{Pred} \leftarrow \sum_n \sum_k w_{kn}^{LMS^n} B_k^n$
  $\Delta s_i^{Pred} \leftarrow \Delta s_i^{Pred} + \delta s_i^{Pred}$
end loop

The internal mechanisms of the SLE of the sensor i are shown in **Algorithm 1**. Here, $\mathbf{S}$ is the vector of values s_i of all sensors, and $\mathbf{M}$ is the vector of values m_n of all motors. The SLE is performing a prediction s_i^{Pred} of the variation of a sensor i. The predictor is initialised with a short term memory of the real value, s_i^{Stored}, which takes the current value s_i^{Real}. Given an arbitrary threshold Th_1^{Step}, chosen empirically, when the sensor moves enough, i.e., $|\Delta s_i^{Real}| > Th_1^{Step}$, or the predicted value moves enough, i.e., $|\Delta s_i^{Pred}| > Th_1^{Step}$, the SLE evaluates the goodness of the prediction given another smaller arbitrary threshold which we empirically defined as $Th_1^{Err} = \frac{Th_1^{Step}}{2}$. Then, the predictor is re-initialised. Considering that learning requires that the robot has only one motor activated at a time, named na, if the prediction is found not good enough, then the SLE recruits a new neuron in the SAW^{na} (related to the activated motor na) and learns the subsequent weight in the LMS^{na} (also related to the same motor na). This method allows it to predict the real observed variation Δs_i^{Real}. The learning is contextualised by the current sensor configuration $\mathbf{S}$ and is linear to the accumulation of motor magnitude M_m of the motor n since the last initialisation. At each iteration, the motor magnitude and the predicted sensor variation are accumulated until the next initialisation step.

B. Using basic skills : the Sensorimotor Law Simulator

The SLE is the only learning mechanism in DSM, which simply learns predictive sensorimotor laws of covariation between sensors and motors. In turn, this predictive mechanism is used to provide actions [81], independently from the learning process. Since no inverse model is learned, the problem of learning one unique solution among an infinite set (see Sec. II-B), is successfully tackled. Equally, the problem of learning multiple inverse models that are found to be useful only in some contexts, rendering them difficult to be learned and to be selected, is avoided (see [39]).

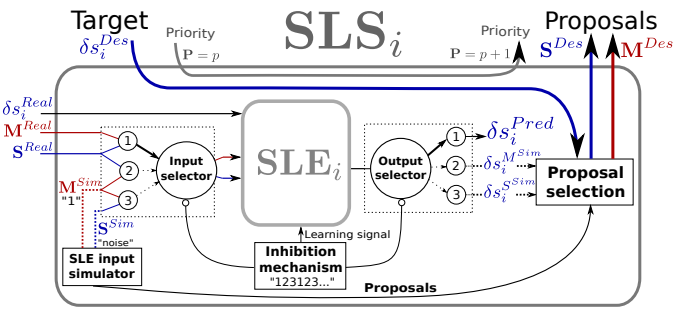


Fig. 2. Sensorimotor Law Simulator for sensor i. An inhibition mechanism allows three various inputs to be sent to an SLE, coming from either real inputs, simulated motor inputs or both sensor and motor simulated inputs. For a given target, the SLS computes the desired sub-goals $\mathbf{S}$ and motor commands M, based on results of the simulations (notice that no learning is involved).

In DSM, an action is selected based on an affordance mechanism. The idea is that the inversion of the learned action-effect associations (theoretically possible according to the ideomotor principle) leads to several action possibilities, i.e., the affordances. Here, we deem affordances as a desire to apply the sensory variations offered by some simulated inputs of the SLE. Hence, the simple observation of the world triggers affordances, which are then used for goal oriented actions.

In line with Moller *et al.* [64], we define affordances as being the predictions computed by the SLE when motor inputs $\mathbf{M}$ match $\mathbf{M}^{Sim}$, a vector generated by the simulator, and not the real vector $\mathbf{M}^{Real}$. Since the SLE explores the local linearity between the predicted sensor and motor inputs and given the vector $\mathbf{M}^{Sim}$, it is possible to compute the precise motor magnitude in order to achieve a goal *in the current local context*.

Extending this work further, we propose the use of the same simulation mechanism with the other input of the SLE being utilised, i.e., a simulated sensor input $\mathbf{S}^{Sim}$. Now, the result of the prediction is not an immediate affordance, where acting in the current context triggers the predicted output. It can be described as a "context-affordance" or a delayed one. That is, an affordance of a context that leads to an immediate affordance. In other words, a prediction based on $\mathbf{S}^{Sim}$ can be used to trigger a desired sub-goal $\mathbf{S}^{Des}$, that, when achieved, it allows the robot to experience a previously simulated context and access any further actions associated with it.

Instead of learning an inverse model, we propose a Sensorimotor Law Simulator (SLS), depicted in fig. 2. Most of the mechanisms involved in SLS are similar to those described by Pezzulo and Castelfranchi in [7] (more specifically in stage 2). First, we propose an inhibition mechanism that facilitates the use of multiple input vectors with SLE. Inhibitory mechanisms are found to be common in the brain. As Pezzulo and Castelfranchi suggest, inhibitory mechanisms can act as switches. Nevertheless, as the in-depth development of a complex inhibitory mechanism is out of the scope of

this work, a simple switching mechanism capable of offering combinations of different inputs, is employed. As mentioned in Sec. III-A, the considered inputs for making a prediction are the sensor and motor vectors $\mathbf{S}$ and $\mathbf{M}$, respectively.

The simulator populates the simulated sensor and motor vectors $\mathbf{S}^{Sim}$ and $\mathbf{M}^{Sim}$. Due to the local linearity of the learning, non-zero values (i.e., 1) are sufficient for all elements in the vector $\mathbf{M}^{Sim}$, whereas the sensor vector $\mathbf{S}^{Sim}$ is arbitrarily filled with noise. The inhibitory mechanism in SLS switches between the following three input combinations (depicted also in fig. 2):

① Using real inputs $\mathbf{M}^{Real}$ and $\mathbf{S}^{Real}$. This is used for learning. The output δs_i^{Pred} is the predicted sensory variation of sensor i.

② Using real sensors $\mathbf{S}^{Real}$ but simulated motors $\mathbf{M}^{Sim}$. The output is the so-called "affordance" value $\delta s_i^{\mathbf{M}^{Sim}}$, and corresponds to the sensory variation of the sensor i, assuming the robot executed the motor command $\mathbf{M}^{Sim}$. Note that a high value of $\delta s_i^{\mathbf{M}^{Sim}}$ indicates a high correlation between the corresponding motor and the sensor i, within the current context. Later in the paper, this value is reflected to an efficiency parameter E.

③ Using simulated inputs $\mathbf{M}^{Sim}$ and $\mathbf{S}^{Sim}$. The output $\delta s_i^{\mathbf{S}^{Sim}}$ reflects the suitability of the simulated context (i.e., the "context-affordance"), that is, a context in which the motor command $\mathbf{M}^{Sim}$ triggers the desired sensory variation to the sensor i.

Considering the two simulated outputs given by ② and ③, a mechanism termed Proposal Selection determines the simulated $\mathbf{M}^{Sim}$ and $\mathbf{S}^{Sim}$ relevant for the desired δs_i^{Des}. This is seen as a type of affordances' competition but between immediate and delayed, "context-affordances". In the implementation used in this work, only one desired output is sent (i.e., $\mathbf{M}^{Des}$ or $\mathbf{S}^{Des}$). If the immediate affordance using $\mathbf{M}^{Sim}$ is strong enough, then a motor command $\mathbf{M}^{Des}$ is desired. Otherwise, if the delayed affordance using $\mathbf{S}^{Sim}$ is strong enough, a sub-goal $\mathbf{S}^{Des}$ is desired. A threshold Th_2 is used to determine whether the affordance is strong enough. More specifically, when this threshold is too low, the corresponding desired motor command is always triggered. On the contrary, when the threshold is too high, neither the corresponding desired motor command nor the sensor pattern are triggered.

On the first hand, if $\delta s_i^{\mathbf{M}^{Sim}}$ is above the threshold Th_2 (i.e., if the efficiency E is high enough), we then compute and send the appropriate motor command, using the local linearity between sensors and motors as shown below:

$$M^{Des} = \frac{\delta s_i^{Des}}{\delta s_i^{\mathbf{M}^{Sim}}} \tag{6}$$

This motor command may become inaccurate as a result of inaccurate learning or due to changes to the contexts. However, M^{Des} is updated at each iteration step and is continuously adapted.

On the other hand, if $\delta s_i^{\mathbf{M}^{Sim}}$ is less than the threshold Th_2, the model tests the efficiency of the context that gives the highest delayed affordance. At each iteration, the proposed selection mechanism checks whether $\delta s_i^{\mathbf{S}^{Sim}_{Max}}$ has the highest value ever encountered. If a higher value is experienced instead, it updates the couple formed by $\{\mathbf{S}^{Sim}_{Max}, \delta s_i^{\mathbf{S}^{Sim}_{Max}}\}$ with the current values $\{\mathbf{S}^{Sim}, \delta s_i^{\mathbf{S}^{Sim}}\}$. Then, if $\delta s_i^{\mathbf{S}^{Sim}_{Max}}$ exceeds the threshold Th_2, the corresponding simulated input is sent; $\mathbf{S}^{Des} = \mathbf{S}^{Sim}_{Max}$. The flowchart related to SLS in fig. 4 illustrates those internal mechanisms of DSM.

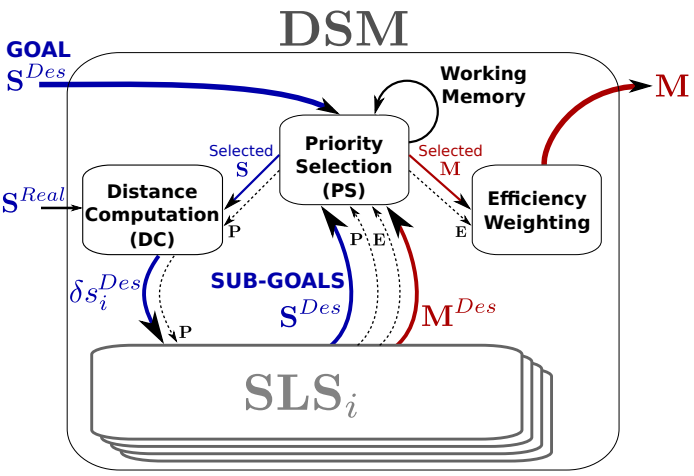


Fig. 3. The Dynamic Sensorimotor Model. For an initial given goal, DSM generates sub-goals and selects the most efficient motor commands, resulting from multiple SLSs. It employs a selection mechanism (depicted as priority mechanism PS) in order to choose between sub-goals and the initial goal, when the latter is directly feasible. It then sends the desired variation of each sensors for achieving the selected goal to the corresponding SLSs. A working memory is in use to keep track of the selected goal.

C. Dynamic Sensorimotor Model architecture

The Dynamic Sensorimotor Model deals with various desired goals and motor commands. The idea is that for one goal, DSM uses multiple SLSs in order to find the most appropriate motor commands and sub-goals and allow the robot to act by implementing sequences of actions that have not been learned, per se, in the past.

A goal is a vector $\mathbf{S}^{Des}$ of l sensors, with l being less or equal to the total number of sensors in the system. The distance between the current sensor values $\mathbf{S}^{Real}$ and $\mathbf{S}^{Des}$ reflects the distance between the actual and desired contexts, and is calculated in order to obtain l different δs_i^{Des} that are then delivered to the l corresponding SLS_i, see fig. 3.

The latter send sub-goals $\mathbf{S}^{Des}$ or motor commands $\mathbf{M}^{Des}$. Here, a selection mechanism is required to deal with the selection of goals and recursive sub-goals, and with the various motor commands that might happen to control the same robotic motors. In our implementation, we propose a priority mechanism where each goal (or sub-goal) is coupled with a priority value P. For the initial goal, $P^{Goal} = 1$. The mechanism selects the priority with the maximum value and sends the selected $\mathbf{S}^{Des}$ and $\mathbf{M}^{Des}$. The $\mathbf{S}^{Des}$ coupled with its priority $P = p$ is then sent to the distance computation mechanism. The couple is then received by all related SLSs, which eventually send $\mathbf{S}^{Des}$ or $\mathbf{M}^{Des}$ coupled with a priority $P = p + 1$.

Once the priority selection mechanism has completed, it is still possible for motor commands to have values that negate

each other on the same motors since each SLS can send $\mathbf{M}^{Des}$ on all values with the same priority. To overcome this, another selection mechanism for all the $\mathbf{M}^{Des}$, called the efficiency mechanism, is used. Motor commands coming from SLSs are individually coupled with efficiency values E. In terms of calculating E, one can argue that the value $\delta s_i^{\mathbf{M}^{Sim}}$ (see Sec. III-A) that corresponds to the affordance value, can be enough. However, it does correspond to the immediate efficiency and is not computed with respect to the "usefulness" of this affordance. In this work, we propose that $E = \delta s_i^{\mathbf{M}^{Sim}} * \delta s_i^{Des}$, so the efficiency finally becomes related to the potential amount of work that needs to be done in order to achieve the current goal. As E is strongly related to the affordances in the current context, a selection method based on E values is close to the idea of affordance competition (see [61]). Although there exist different ways to compute the motor commands $\mathbf{M}$ values based on the efficiency E, such as linear combinations, etc., in this work the winner motor command is the one with the maximum efficiency. This is to ensure precision in the desired motor magnitudes in the system.

In fig. 4, a flowchart of a proposed implementation of the DSM is shown, including the main equations that are used for the internal mechanisms. Notice that for the sake of simplicity, the flowchart has no efficiency weighting mechanism.

Goals, i.e., pairs of a vector of desired sensor pattern and priority values, are sent to the priority selection module which in turn selects the goal with the highest priority. The distance computation module then sends the required sensor variations, coupled with their priority, to the corresponding SLS. The latter then uses simulations within its SLE in order to compute the sensor variation related to a motor movement in the current or in another context. Given a threshold, it sends either a motor command or a desired context (i.e., a sensor pattern) as a new sub-goal with a higher priority.

DSM allows motor commands to be performed as soon as possible, with sequential behaviours emerging on the fly. For instance, if the goal consists of reaching to a given point, the SLS corresponding to the hand sends motor commands $\mathbf{M}^{Des}$ to get the hand closer to the target (it sends no $\mathbf{S}^{Des}$ at this point). If the hand cannot get close enough to the target, the SLS sends a $\mathbf{S}^{Des}$ to have the tool in the hand, considering the tool is perceived as an extension of the arm. It corresponds to a sub-goal "having the tool in the hand". Then, through implementing commands from other SLSs (see next section for more details on this experiment), the hand's SLS can be assigned a new δs_i^{Des} in order to reach towards the tool. Here, the following problem is encountered. After the new assignment, the previous sub-goal "having the tool in the hand" is no longer sent, rather it is somehow forgotten. The problem originates from the fact that in this particular sequence hierarchy, the SLS of the hand is found twice in the hierarchy of goals. Pezzulo and Castelfranchi suggest that a working memory is needed in order to maintain distal goals [7]. Adapting their suggestion, all s_i^{Des} of the vector $\mathbf{S}^{Des}$ are stored in a working memory within the priority selection module, which is also responsible for pruning them, one by one, when $|s_i^{Des} - s_i^{Real}| < Th_2$.

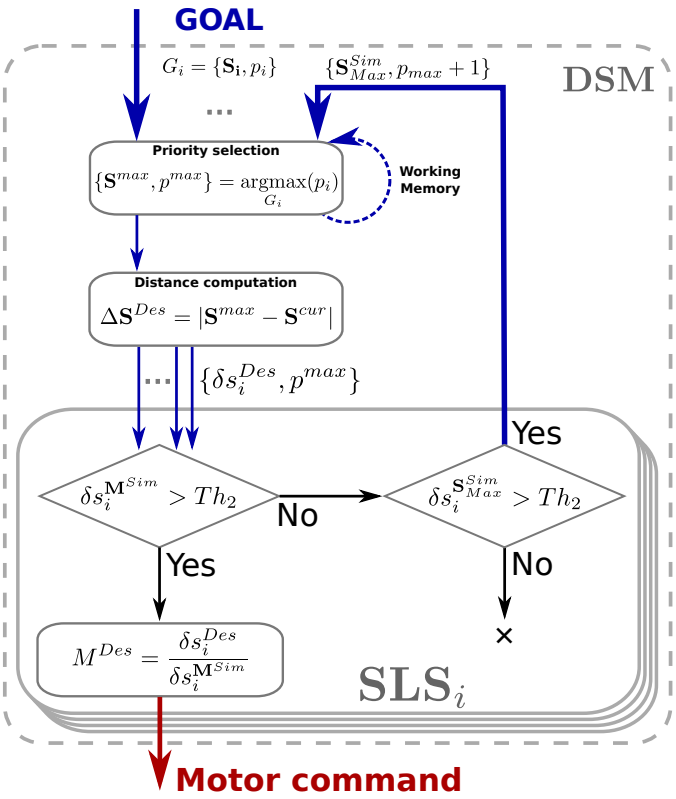


Fig. 4. Flowchart of the implementation of DSM used in this work.

IV. Experiments

A. Simulation experiment: SLE performances

In previous works, we demonstrate that the SLE is able to efficiently predict sensory variations and can be used for reaching tasks. In [33], we simulate a four degree of freedom (DoF) Katana arm to test the SLE's learning performance for reaching tasks in a 3D space. We also compare its performance against an absolute and a control strategy model. Each strategy performs a motor babbling that explores random arm configurations and redundancies. On a regular basis, we test the reaching performance; a continuous sequence of target coordinates in 3D space has to be reached by the hand of the Katana arm, and the mean distance in meters from the hand to the target is computed during the whole task. The same sequence of target is repeated for each test. The results show that the model based on SLEs is the fastest to decrease. In fig. 5 the mean error curve at different steps of the development is seen. A comparison with other models and results on performances when the body schema is changed through a tool is found in [33].

B. Real robot experiment: DSM on the fly sequences

A real robot is used in order to evaluate the efficiency of DSM and its ability to make sequences of actions on the fly. The aim is to maximise a reward signal, by reaching a given target with the hand of a robot. Two different tools are available in the experiment, with different affordances. They both have the property to extend the body schema,

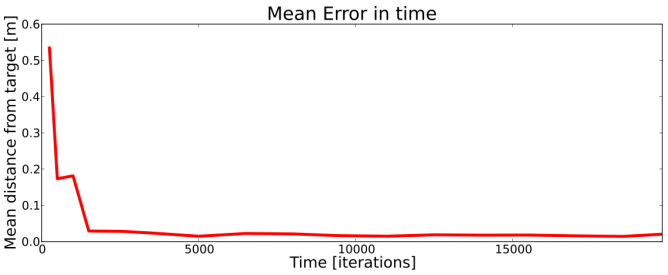


Fig. 5. Mean error of a sequence of reaching tasks repeated at different step of development, in simulation with a 4-DoF Katana arm. Between each trials, a babbling phase is performed for learning with the three SLE for axes X, Y and Z.

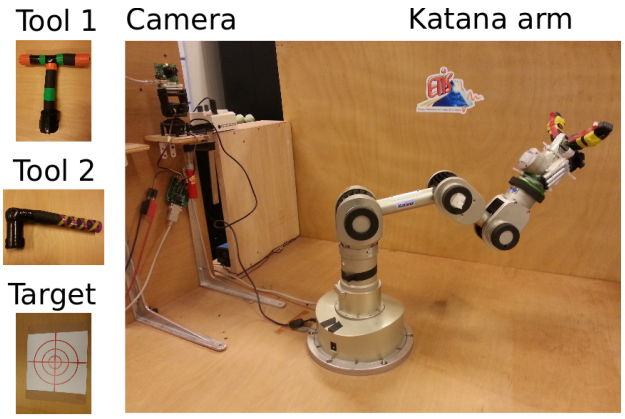


Fig. 6. Experimental setup.

i.e., geometry of the arm, but not in the same direction. Tool 1 extends the arm vertically whereas tool 2 extends it horizontally. In order to show various functions of those tools, we block the articulation of the arm so both tools can be useful in different situations for a reaching task. Tools are useless when the target is already reachable, but tool 1 becomes useful when the target is out of reach on the Y axis, while tool 2 becomes useful when the target is out of reach on the X axis. Basic sensorimotor rules are learned through different SLEs and we see how our model is able to maximise its reward thanks to the ability of DSM to generate and utilise sequences.

Apart from the Katana arm and its hand, the experimental set up consists of a camera, two tools and a target, see fig. 6. The camera is used to recognise and localise the arm, the tools and the target on the image. This is achieved by the use of a previously developed bio-inspired object recognition algorithm, based on local points of interests (see [82]). It gives the robot a 12-neuron vector, 4 for the presence of the different objects and 8 for their positions on the X and Y axes. In this experiment, the target is out of reach hence the robot is expected to reach towards the tool and ultimately grasp it. When the tool is grasped by the robotic hand, the robot perceives its hand position as being extended to the tool position. A reward signal is also given to the robot. This signal increases as the perceived distance from the hand to the target $s_{H \to Target}$ decreases.

We determine the values of the two DSM thresholds empirically. That is, $Th_1 = 0.1$ and $Th_2 = 0.05$ as in our implementation sensors vary globally between 0 and 1.

We make use of three SLEs. Each of them is dedicated to the prediction of one sensor variation and receives sensors and motors as its input (see fig. 1):

- SLE_{H_x} predicts the variation of the perceived position of the hand on the X axis. It takes the proprioception θ_1 of the first degree of freedom as its sensor input. Since the Katana arm has its own PID, we do not have direct control of the motor command. The variation of the proprioception $\delta\theta_1$ is then used as a motor input. Another couple of sensor and motor is used to predict the variation induced by tool 2. It takes the distance $s_{H \to Tool2}$ from the hand to the tool 2 as a sensor input, and the grasping motor command δ_{Grasp} as a motor input.
- SLE_{H_y} predicts the variation of the perceived position of the hand on the Y axis. It takes the proprioception θ_2 of the second degree of freedom as its sensor input. For the same reason stated above, $\delta\theta_2$ is the motor input. Another couple of sensor and motor is also used for this SLE to predict the variation induced by tool 1. The distance $s_{H \to Tool1}$ from the hand to the tool is used as a sensor input, and the grasping motor command δ_{Grasp} as a motor input.
- SLE_{Reward} predicts the variation of the reward signal. Using the common coding property, we use as motor input the variation of the perceived distance between the hand and the target $\delta s_{H \to Target}$. This distance is directly linear with the reward and is not context dependant, so no context is needed (notice that one linear law is enough). Nevertheless, since a SLE needs a sensor input we also take the distance $\delta s_{H \to Target}$ as the sensor input.

In the whole DSM architecture, the SLEs are the only learning modules. They learn independently from each other and regardless of any future task. Each SLE is associated with a corresponding SLS that runs in parallel and also independently from others. The learning of the three SLEs is then performed.

No major differences are reported when running DSM on a real robotic platform as compared to the simulations. However, due to the lack of explored redundancies and the noisy visual input streams (see estimated positions of the hand while performing smooth movements in fig. 7 and 8), meaningful threshold values have to be higher and, as a result, the system becomes less sensitive to capturing a rich set of sensorimotor laws. Although the system is capable of learning, it is found to acquire just a few minimal rules about covariations between proprioception and visual positions. Importantly, both SLEs of the hand are able to learn a context, i.e., a proprioception area on θ_1 or θ_2 encoded in the SAW) where the hand could not move more (supposing articulations limits).

Note that even if all SLEs have to compute their predictions at each iteration (which is computationally heavy), they learn independently from each other and in our implementation they run in parallel and use asynchronous communications.

Here we describe sequence of events of the experiment. The only input, in DSM, is an external goal concerning the reward (see fig. 3).

Phase 1: Reaching the target. We give to DSM the goal $Reward = 1$. Being an external goal, it is associated with the priority 1 ($P = 1$) in the priority selection (**PS**) module. The distance from the current value of the reward value to 1 is computed by the Distance Computation (**DC**) module and it is sent to SLS_{Reward}. This SLS sends a motor command $\delta s_{H \rightarrow Target}$ to **PS**, associated with the $P = 2$, which is interpreted as putting the hand on the target. This is done at each iteration of the DSM model. **DC** computes distances $\delta s_{H \rightarrow Target}$ on X and Y axis and sends them to SLS_{H_x} and SLS_{H_y}, associating them with $P = 2$. Both SLS then send motor commands on θ_1 and θ_2, associated with $P = 3$. Here, no other motor commands are sent so the Efficiency Weighting module is not necessary. A reaching behaviour is then observed.

Since the SLE has learned that there is a point where $\delta s^{Pred}_{H_y} < Th_2$, when a target is out of reach because it is on top of the hand in the Y axis, the SLS checks if a $\mathbf{S}^{Sim}$ exists where $\delta s^{\mathbf{S}^{Sim}}_{H_y} > Th_2$ (see III-C). If no rule associated with tool 1 has been learned by SLE_{H_y}, the arm simply does not move and the robot stays in *phase 1*. If a rule associated with the tool 1 that extends the arm, hence causes a variation on the Y axes, has been learned (note that it can happen at any time, independently of other learning or behaviours), then $\delta s^{\mathbf{S}^{Sim}}_{H_y}$ can have a value beyond the threshold. Typically this holds when $\mathbf{S}^{Sim}$ corresponds to having the hand close enough to the tool 1. The same mechanism applies with SLS_{H_x} and tool 2.

Phase 2: Reaching of the tool. What we describe here applies for both tool 1 and tool 2. In this phase, the target is not reachable because it is too far from the hand in the Y axis, and $\delta s^{\mathbf{S}^{Sim}}_{H_y} > Th_2$ in the SLS_{H_y}. The SLS sends a sub-goal on $s_{H \rightarrow Tool1}$ associated with $P = 3$. This time, **PS** receives 2 desired values for H_x and H_y, since it had also received $\delta s_{H \rightarrow Target}$ with $P = 2$ from SLS_{Reward}. The desired value with the highest priority is then selected: **DC** computes the distance from the hand to the tool 1, and sends it to SLS_{H_x} and SLS_{H_y}, associating it with $P = 3$. As in *phase 1*, the SLS sends motor commands and the hand tries to reach tool 1. Note that unlike *phase 1*, **PS** does not receive $s_{H \rightarrow Tool1}$ at each iteration. Instead, it is received only once, since after setting a sub-goal SLS_{H_y} just sends motor commands. This justifies the need of a **working memory** in **PS**, in order to keep this sub-goal in memory until it is met.

Phase 3: Reaching of the target with the tool in the hand. The hand grasps the tool and the arm is perceived as being extended. The hand is now perceived at the tool position on the X and Y axes. The sub-goal is erased from the **working memory** in **PS**. The system then behaves as in *phase 1*.

On fig. 7 and fig. 8, we present results of experiments where the targets are unreachable along the Y and X axes. In both cases, the task is the same, i.e., increasing the robot's signal reward. Both tools present to the robot during the whole experiment. The results show the trajectory of the arm as perceived by the object recognition algorithm. We can observe the 3 phases described above; i) the robot tries to reach the target, ii) it reaches a point in its proprioception space where

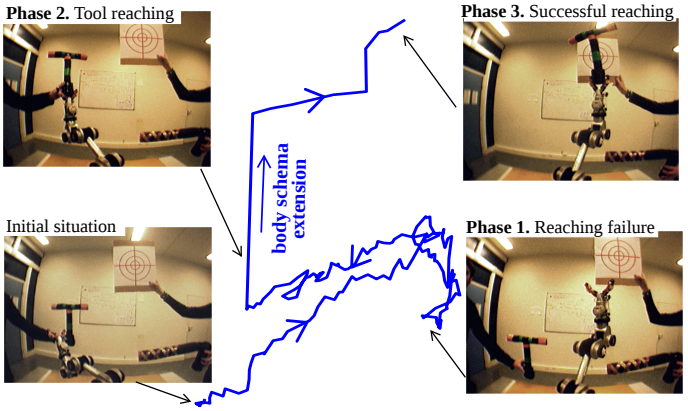


Fig. 7. Hand trajectory of the Katana arm (as perceived by the object recognition algorithm) with a target out of reach along the Y axis. The robot makes a detour to grasp the adapted tool 1.

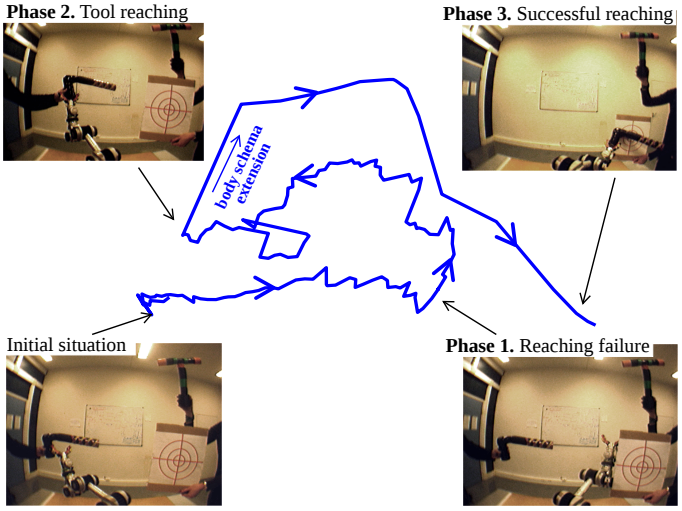


Fig. 8. Hand trajectory of the Katana arm (as perceived by the object recognition algorithm) with a target out of reach along the X axis. The robot makes a detour to grasp the adapted tool 2.

it move further along the X or Y axis, and iii) it makes a detour on-the-fly in order to select the tool, most suitable for its problem.

In fig. 9, we show more detailed results of an experiment performed with just tool 1. In a), the positions of the hand (in blue), the tool (in green and dotted) and the target (in red and dashed) are depicted for both X and Y axes, during the course of the experiment. In b), motor commands given by θ_1 and θ_2 are in dashed blue and green, respectively. In c), we show the priority values received by the SLS_{Reward} in red, by the SLS_{H_x} in dashed blue and by the SLS_{H_y} in dashed green. During the *phase 1*, the SLS_{Reward} sends desired values for the hand in order to reach the target, with priority $P = 1$. Those values are sent to both SLS_{H_x} and SLS_{H_y} which in return send desired motor commands θ_1 and θ_2 with priority $P = 1+1 = 2$. In fig. 9 a), we observe that the hand (in blue) is trying to reach the target (in red).

In terms of the second phase, the target is out of reach on the Y axis and the arm is within the proprioception area where no further upward movements are possible (previously

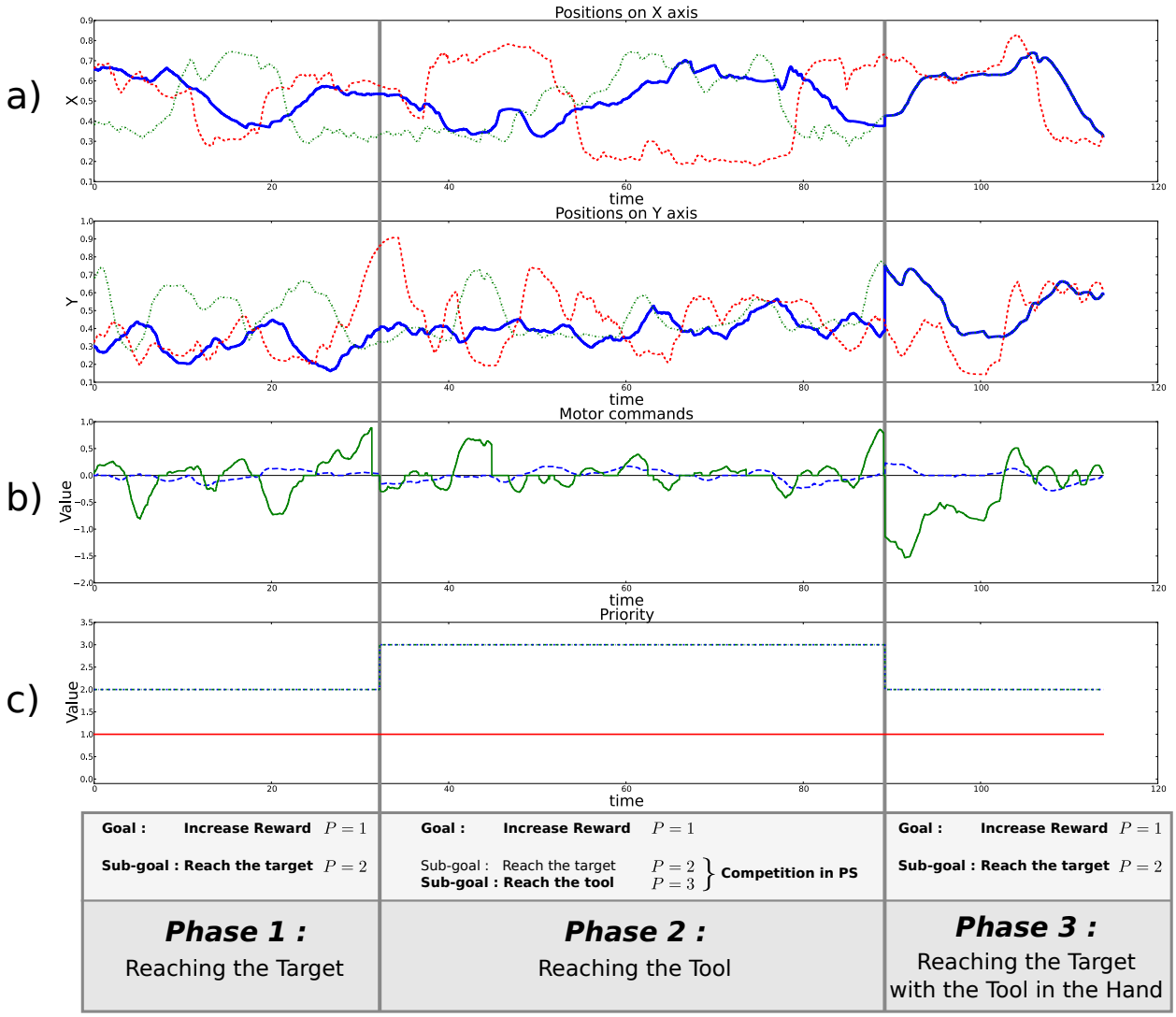


Fig. 9. In a), the hand appears in blue, the tool in dotted green and the target is depicted as dashed red on the X and Y axis, as a function of time (in seconds of the experiment). In b), motor commands given by θ_1 and θ_2 are depicted as dashed blue and green respectively. In c), we see the priority values received by the SLS_{Reward} in red, the SLS_{H_x} in dotted blue and the SLS_{H_y} in dotted green.

learned by the SLE): $\delta s_{H_y}^{Pred} < Th_2$. Due to the eq. 6, the motor commands on θ_2 is 0 at this exact moment (see fig. 9 b) in green). The SLS_{H_y} then sends a sub-goal on $s_{H \rightarrow Tool}$ associating it with $P = 2 + 1 = 3$. Moreover, SLS_{H_x} and SLS_{H_y} send motor commands θ_1 and θ_2 in order to reach the tool with priority $P = 3$. In fig. 9 a), we can observe that the hand (in blue) is trying to reach the tool (in green).

Once the tool is grasped, the sub-goal is erased from the working memory and like in phase one, the desired value coming from SLS_{Reward} is selected again. The robot tries to reach the target but this time with the tool in its hand. In fig. 9 a), we observe that the hand (in blue) and the tool (in green) have merged, and are trying to reach the target (in red).

The delay between motor commands and the hand position is caused by the object recognition algorithm. The difference in amplitude between the two motor commands is due to the sensorimotor laws learned by SLS_{H_x} and SLS_{H_y}. The efficiency is higher on the X axis with our Katana arm, so less motor magnitude is needed in order to move on this axis. Note that the PID of the arm changes the real command sent to the effector.

V. Discussion and Conclusion

As mentioned before, the only learning mechanism in DSM is in fact the SLE, dedicated to making predictions related to sensory variations. The SLS, through its simulation and inhibition mechanisms, uses SLE's predictions to perceive affordances. The affordances are ultimately used to determine what should be performed in order to fulfill a given goal. Motor simulation leads to desired motor commands, while sensor simulations lead to desired sensory patterns, i.e., sub-goals. Desired motor and sensor values are either inhibited or taken into account, thanks to a priority mechanism. Temporary inhibition of previous goals enables the robot to perform intermediate tasks and to generate sequences of them. A working memory is responsible for storing sub-goals without depending on the SLS, allowing the latter to send only desired motor commands or sensory patterns in connection to the

current desired shift. Consequently, each couple of SLE and SLS is unaware of the current hierarchy or sequence of event, even though they both result from what the SLE has previously learned. Even with lack of planning, this approach allows a cascade of sub-goals to occur and to be calculated on-the-fly. A sequence of various sub-goals may occur depending on the experiment, as we illustrated with the experiments of two different tools. Moreover, such sequences may occur even if they were neither learned nor observed before, as in our experiments.

Sequences depend entirely on what each SLE has learned during the robot development, and on the situations it has faced. With our approach, sequences are not meant to be put at the top of existing elementary behaviours, rather they are the direct consequences of specific low-level sensorimotor laws. For instance, in our experiment the detour to the tool is due to the way the robot has learned the law of the tool, that is, the robot must have the capacity to learn in a continuous fashion, and to be able to change previously learned laws.

A. Comparison with related approaches

Tool-use raises the problem of kinematic adaptation. In [83], a single recurrent neural network is used to learn the inverse kinematics of different tools, with no relearning or forgetting of the body schema. However, the length of the tool is a parameter of this neural network and must be known in advance. Considering this problem, in [84] Jamone et al. show how the kinematics could be adapted to tool-use when no information is given about the tool at hand. They used the IMLE model which allows the incremental learning of different solutions to similar input locations, and thus represents the relations from an input vector to an output vector in specific unknown contexts. During both the learning and the using phase, the algorithm automatically deals with an appropriate set of local linear models and this knowledge is exploited for a reaching task.

However, Nabeshima et al. [85] claim that the real problem might actually be related to the detection of the body alterations based on the sensory data, and they propose a model for adapting the body schema of the robot with a tool. The robot (a 2-joint planar arm and a camera) learns the mapping between the spatial and temporal integration of visual and tactile information when the hand of the robot hits a target. The mapping is then stored in an associative memory. When the robot touches the target with a tool (a stick), a new mapping of visual and tactile information is learned in the associative memory, and the kinematics controller of the arm is adapted.

Although preliminary results are given in [33], the kinematic adaptation during tool-use is out of the scope of this paper. However, this is considered as a subject for further investigation by using a new categorisation layer. Concerning the problem raised by Nabeshima et al., the learning process of the SLE is dedicated to the detection of alterations of the predictions (and, in case of body predictions like the movement of the hand, of the body) based on sensory data. This detection relies on the accumulated error during the duration of a movement (until δs is beyond a threshold). However, the main difference is seen when we focus on how such a knowledge can be used to change the behaviour of the robot in a reaching task, by making a self-generated detour.

In [86], a dynamic neural field (DNF, see [87]) method is proposed to bridge the gap between the lower level sensorimotor dynamics and the higher cognitive processes, as in planning. The authors propose elementary behaviours such as “look for an object with colour X” or “move arm to position Y”, associated with DNF conditioning initiation and termination (condition of satisfaction) for those behaviours. Each behaviour is also associated with a motivation field responsible for triggering it and a precondition field which defines the conditions that need to be met. Once the motivation field is triggered, the DNF of initiation is triggered but can also be inhibited, in case that the precondition field is not satisfied. In this case, the precondition field can trigger the motivation field that belongs to other elementary behaviour. Subsequently, once its condition of satisfaction field is met, it stops inhibiting the precondition field of the first elementary behaviour.

This mechanism is close to the one described in DSM. The categorisation layer in the SLE is used (via simulations) to send the desired sensory pattern (equivalent to a motivation), when the motor commands are not efficient enough. This is seen in cases when there is no affordance, i.e., an equivalent to the precondition field. Similarly, when the desired sensory pattern is reached (equivalent to the condition of satisfaction), a motor command is executed (equivalent to the initiation field).

The two approaches differ at the sub-task initiation or termination. In DSM, the equivalent to the condition of satisfaction is not given by the sub-task, but by the SLS which initially expresses a desire for it. The problem of initiation and termination of a sub-task is not related to a behavioural achievement, but to the learned sensorimotor laws. More precisely, it depends on the categorisation layer of the SLE. Thus, the request is not made for a behaviour but directly for a low-level sensory pattern. Our approach is designed to directly bridge low-level sensorimotor skills to higher level sequence abilities, boosting the latter’s characteristic of being very plastic. In fact, all SLS can send desired sensory patterns and create a specific order of sequence on-the-fly, without doing any planning. Due to a working memory, remote to SLS, they are able to indirectly change their own pursued goals, as shown in tool-use experiments.

In [88], the authors describe a model that learns associations between parameterised motor skills and their corresponding parameterised tasks, based on a measure of competence progress. They propose a way to optimise the ability of the robot in solving tasks, based on artificial curiosity. In their work, the robot explores the task space instead of performing a motor babbling. However, their approach does not focus on the ability to solve new tasks, or to solve them in a new manner.

In [89], a set of closed-loop dynamic controllers is linked to a finite state machine whose states activate one or multiple controllers. Each controller is driven by a primitive sensorimotor potential function. State transitions are triggered by the convergence of each controller to a stable attractor. These transitions are learned through Q-learning using a intrinsic

motivation reward that detects the convergence (called quiescence) to such attractors. As a result, the system is able to uncover affordances defined by the specific sensory areas where each behaviour is stable, and can generate complex sequences by following the learned transition rules guided by controllers convergence. Note that this approach does not directly predict the sensory consequences of actions, but rather it exploits criteria related to behavioural stability in order to drive controllers' activation to achieve a particular task.

In [90] the authors investigate open-ended exploratory systems and goal-finding behaviours, and specifically choose motor babbling behaviour (or even playing behaviour) to drive learning instead of goal-driven methods. The authors are interested in transitions between multiple stages of behaviour. They show how selected constraints on the robot, coming from developmental psychology, help reducing the complexity of learning at each stage of development. Their results also illustrate how the learning acquired during previous stages is used to scaffold the learning during the later stages. However, the learning is based on sensor and motors pairs, a method that is discussed in the introduction as being an absolute learning strategy.

In [91], Perotto proposed a model (CALM) where regularities are learned in schemas composed of three vectors: context + action $\rightarrow$ expectation. Each vector is composed of discrete values, in contrast with the work proposed in this paper, where the context, the motor and the expected sensor variation are continuous values. However, those states can contain undefined values, which offer a good generalisation property. The learning is incremental, and schemas can go through several operations such as they can produce more specific ones, they can be corrected after a novel experiment, or they can even fused with old schemas when necessary. Moreover, values are not necessarily observable and can be "synthetic", in order to deal with partially unobservable environments or abstract useful properties (also designed to bridge the gap between high-level symbolic properties), or even to mark different steps in a sequence of action. In DSM, such synthetic or hidden sensors or motors may be useful when dealing with partially unobservable and deterministic environments. Although the CALM model is able to predict final states, it does not predict changes which would offer a better generalisation property.

In [92], Ugur et. al. use affordance learning for plannification. In their work, the robot learns associations between the initial features of object perceived and the effects perceived after a given behaviour, through a babbling phase. The result is effectively the difference between the final and the initial object's features. Since the final states can be computed by adding the effect of previous to the current state, it is possible to create a search tree with nodes that are composed of perceptual states, and edges which represent behaviour-object pairs. Thus, for any given goal the robot creates such a tree and gradually expands it by starting at where a state has a minimal distance to the goal, and then selects a sequence from the tree when a leaf node leads to a state which satisfies the goal.

This approach can be considered as close to the work presented in this paper, since the robot learns associations between continuous sensory contexts and effects through actions, and discovers sequences that minimise the distance to the goal. However, there are important differences that set our work apart.

Initially, in DSM, every input (sensors and motors) and output (sensor variation) of the SLE is continuous. Motor commands are not discrete, nor defined in advance as precoded behaviours. The motor magnitudes are either low-level commands or predictable sensor variations, thanks to the hierarchical property. Next, the categorisation performed in DSM is not determined by the sensor space and topology, but it is triggered by the prediction error of the sensorimotor laws during babbling (see Sec. III-A). Hence, the categorisation depends on the action performed and not purely on the sensors. Furthermore, categories are just used to contextualize sensorimotor laws. Next, the effect space (sensor derivative) and motor spaces (motor magnitudes) are not categorised, but linearly associated. This linear association is useful when considering the space of simulated contexts as a space wherein motor actions are possible, i.e., the robot can possibly perform actions towards a simulated context in the same way it would act within the real environment. This allows the robot to possibly act on the context space and obtain results that match actions within the real environment.

Finally, in DSM sequences are generated without the use of a dedicated planning module: the described mechanism allows the creation of sub-goals without the use of a forward chaining that is based on discrete states. Planning in DSM directly enjoys low-level motor commands as output, and does not consider any predefined list of behaviours. Sequences are created on-the-fly, and each concerned SLE is able to indirectly take advantage of other SLEs for achieving its subgoal. Unlike [92] and similar methods that depend on forward chaining, DSM's sequential step follow an inverse chaining approach, overcoming any combinatorial exploration issues. It is worth noting that even if our model allows planning to be made only one step ahead, i.e., through the simulation of the next step, there is already a creation of a sequence involving sub-goals. Our future research efforts include the ability to anticipate further steps, allowing for longer planning. That is expected to require an inhibition mechanism in order to use internal close loops of predictions and to anticipate subgoals. In the presented experiments for instance, the ability to anticipate the need of the tool and reach it directly could be possessed, instead of initially trying to reach the goal directly.

Finally, the fact that sensor variations and motor values are jointly encoded, according to the hierarchical property of DSM, SLEs are able to receive sensor variations as motor inputs that allows them to deal with high-level input, and ultimately bridge the gap between low-level and more conceptual or symbolic learning, and in turn, planning.

In [93], Chaput proposes a constructivist model called CLA based on a hierarchical property. As in our model, this hierarchical property arises from the fact that CLA sensorimotor schemas have inputs and outputs of the same representation type. The first layer inputs are raw sensors, whereas the next higher layers use the lower layers as their inputs. Chaput also proposes an interesting mechanism to exploit the hierarchical properties: the receiving layer could be prone to time delays,

which can potentially lead to recurrent structures. Even if when a new layer is created, a higher level layer is more likely to be used as in input to it. Chaput way to tackle this issue is by suggesting a fallback mechanism, by which lower layers can be used as inputs of newly created layer, in case when the robot receives a stimulation associated with an new, unexperienced event. The outcome of Chaput's work is considered as a possible design enhancement for DSM.

"Object Action Complexes" (OAC) are designed to bridge the gap between low-level sensorimotor representations, required for robot perception and control as well as high-level representations that support abstract reasoning and planning [94]. Given discrete states, OAC is designed to learn the final states that follow the execution of a motor program, e.g., hard coded action reflexes, elicited by visual inputs. It is worth noticing that the resulting states contain all available information, that can be either relevant or irrelevant for predictions. The goal of the learning mechanism of OAC is to abstract irrelevant information and in turn generate states that contain only the information that is required to successfully predict the effects of actions. This problem exists in the context of DSM and is similar to the one describes in [94], as all sensors are potentially relevant when predicting a sensor variation. Furthermore, DSM needs to abstract only the necessary inputs, not only for having more accurate predictors, but more importantly to create sub-goals using only relevant desired sensor patterns. In the experiments presented here, the relevant inputs were pre-selected, however, this needs to be done by DSM itself and will be addressed in future work.

B. Properties coming from ideomotor principle

Aligned with the ideomotor principle, the SLE is a proposition for having sensory variations dS being perceived as equivalent to motor magnitudes M (assuming locally linear relationships exist between them). As mentioned in the introduction, sec. , the input motor magnitude of a SLE can be a given sensory variation dS. For instance, in our experiment we used $\delta s_{H \rightarrow Target}$ as a motor input for the SLE_{Reward}. This unique characteristic for DSM highlights to two interesting properties:

First, it allows the learning of the robot to be hierarchical as described by Chaput, i.e., the first SLE has raw sensors as in input, whereas the next higher level SLE can received input from lower level SLEs. In the presented experiments for instance, this feature allows the learning of possibly complex laws between the movement of the arm and the environment, which can be considered high-level but detached from the low-level complexity associated with moving the arm. The latter is in fact a result of the learning of another SLE (i.e., SLE_{H_x} and SLE_{H_y}) that remains beneficial to the whole system. Thanks to this feature, it is possible to start with SLEs for very low-level predictions, and gradually scaffold the learning, step by step, through a hierarchical construction.

Second, since the motor input can be a sensory variation, learning by observation is possible as long as a SLE can make use of it as its motor input. Since new SLEs can be added on-line without changing the mechanism, we propose that, for instance, a sensory variation could be considered as a motor input of a SLE as long as the SLE predicting this sensor is "good" enough. In this scenario, learning by observation could happen when the robot has discovered the sensorimotor laws that enable it to possibly reproduce the observed scene.

Such a design implies that new motors can be added on SLEs and that new SLEs can be created and added on-the-fly. This is feasible as a new couple of SLE and SLS is currently used to add new desired motor commands or sensory patterns to the priority selection mechanism. Adding a new motor in a SLE is also possible, since it consists of adding a new contribution on top of other in the LMS (see fig. 1). This allows incremental learning, where previously acquired skills are not removed as new ones are learned.

In the SLE, the S input should also be flexible in order to accommodate new high-level sensors (in our experiments for instance, the objects' presence and coordinates). Similarly to the CALM model [91], it would also be interesting to consider using sensors that are equivalent to "synthetic" values, in order to deal with partially unobservable and deterministic environments. In the next section we discuss problems related to these input sensors.

C. Current limitations and future works

As previously stated, in our approach a sequence is entirely dependant on the sensorimotor laws that the robot learns during its development, e.g., during its babbling phase when is not necessary that a sequence or a specific task is performed. The sensory inputs of the SLE are used when performing simulations, so any desired sensory pattern sent by a SLS_i incorporates sensory information received by all relevant and irrelevant inputs of the SLE_i (see III-A).

Finally, when the system is mature enough to make sequences (in the DSM) and run simulations (in the SLS), the resulting sequence will only depend on the learning which is performed by the categorisation layer of the SLE. For instance, if the categorisation layer fails to distinguish between having or a tool in the hand or not, then a simulation with any simulated input sensors will also fail. The problem of distinguishing relevant context is highlighted by the fact that, from a developmental perspective, new input sensors can be added on-line, and possibly all sensors may be relevant (especially in the case of tool-use). Currently, if the model is given irrelevant input sensors, they will equally participate to the internal mechanisms of the SLS as the relevant ones, leading to random configurations associated with them. Moreover, since many categorisation layers are required (possibly one per couple of motor and sensor), irrelevant inputs can rapidly increase the computational cost. A potential solution is to have a categorisation layer able to distinguish relevant inputs from irrelevant ones (as in "Object Action Complexes" [94]), such as those that constitute a context that has an impact on the prediction of the SLE. Showing promising preliminary results, such solution will be explored in the future. The aim is to investigate the ability of the robot to act on its contexts and, in turn, test the categorisation it autonomously creates.

In our approach, categorisation leads to specific sequences, which is akin to further problems related to the misuse of

affordances. For instance, affordances are usually linked with the visual properties of objects. Although we did not tackle this problem here, a possible way to deal with it in the context of DSM, is to select appropriate visual primitives as inputs of the categorisation layer of the SLE. Then, by exploiting the direct links between what is categorised and the corresponding affordances, a SLS can send desired values that are related to the primitives, so that any object defined by them can be desired.

For instance, in our experiment if we do not an object recognition algorithm for determining the inputs of the SLE_{H_y}, but instead we use visual primitives able to distinguish a long-shaped object, we could envision that the robot will then desire to reach any object that complies with this particular primitive, allowing the use of various objects as tools. The visual properties of objects are associated with particular ways to grasp them. For instance, if the handle of a cup is on the right, the hand should approach it from the right. Such planning is a problem with gradient descent methods like ours, because in our model the robot tries to reduce the distance from the hand to the targeted object, without any other consideration. However, the ability of DSM to create on-the-fly sequences based on delayed or context-affordances may be combined with the possibility of internal simulation loops in order to solve such problems. This is also discussed in [79], where a mental rehearsal is used to tackle the problem of obstacle avoidance. This is also considered as future work.

ACKNOWLEDGMENT

This work was supported by ROBOTEX project.

We are grateful to Jerome Fellus, Christelle Morawski, Alexandros Giagkos and our reviewers for comments which helped improve the manuscript.

REFERENCES

[1] W. James, "The consciousness of self," *The principles of psychology*, vol. 8, 1890.

[2] W. Prinz, "A common-coding approach to perception and action," in *Relationships between perception and action: Current approaches*. Springer, 1990, pp. 167–201.

[3] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, no. 5504, pp. 599–600, 2001.

[4] C. Prince, N. Helder, and G. Hollich, "Ongoing emergence: A core concept in epigenetic robotics," 2005.

[5] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 2, pp. 265–286, 2007.

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

[7] G. Pezzulo and C. Castelfranchi, "Thinking as the control of imagination: a conceptual framework for goal-directed systems," *Psychological Research PRPF*, vol. 73, no. 4, pp. 559–577, 2009.

[8] L. Smith and M. Gasser, "The development of embodied cognition: Six lessons from babies," *Artificial life*, vol. 11, no. 1-2, pp. 13–29, 2005.

[9] D. C. Dennett, "Current issues in the philosophy of mind," *American Philosophical Quarterly*, pp. 249–261, 1978.

[10] D. Pierce and B. J. Kuipers, "Map learning with uninterpreted sensors and effectors," *Artificial Intelligence*, vol. 92, no. 1, pp. 169–227, 1997.

[11] D. Stronger and P. Stone, "Towards autonomous sensor and actuator model induction on a mobile robot," *Connection Science*, vol. 18, no. 2, pp. 97–119, 2006.

[12] L. A. Olsson, C. L. Nehaniv, and D. Polani, "From unknown sensors and actuators to actions grounded in sensorimotor perceptions," *Connection Science*, vol. 18, no. 2, pp. 121–144, 2006.

[13] F. Kaplan and V. V. Hafner, "Mapping the space of skills: An approach for comparing embodied sensorimotor organizations," in *Development and Learning, 2005. Proceedings., The 4th International Conference on*. IEEE, 2005, pp. 129–134.

[14] J. Tani, "Symbols and dynamics in embodied cognition: Revisiting a robot experiment," in *Anticipatory Behavior in Adaptive Learning Systems*. Springer, 2003, pp. 167–178.

[15] Y. Nagai and M. Asada, "Predictive learning of sensorimotor information as a key for cognitive development," in *Proc. of the IROS 2015 Workshop on Sensorimotor Contingencies for Robotics*, 2015.

[16] J. J. Lockman, "A perception-action perspective on tool use development," *Child development*, pp. 137–144, 2000.

[17] F. Guerin, "Learning like a baby: a survey of artificial intelligence approaches," *The Knowledge Engineering Review*, vol. 26, no. 02, pp. 209–236, 2011.

[18] A. Maravita and A. Iriki, "Tools for the body (schema)," *Trends in cognitive sciences*, vol. 8, no. 2, pp. 79–86, 2004.

[19] M. Hoffmann, H. G. Marques, A. Hernandez Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, "Body schema in robotics: a review," *Autonomous Mental Development, IEEE Transactions on*, vol. 2, no. 4, pp. 304–324, 2010.

[20] F. Guerin, N. Kruger, and D. Kraft, "A survey of the ontogeny of tool use: from sensorimotor experience to planning," *Autonomous Mental Development, IEEE Transactions on*, vol. 5, no. 1, pp. 18–45, 2013.

[21] M. E. McCarty, R. K. Clifton, and R. R. Collard, "Problem solving in infancy: the emergence of an action plan." *Developmental psychology*, vol. 35, no. 4, p. 1091, 1999.

[22] S. H. Johnson, "Thinking ahead: the case for motor imagery in prospective judgements of prehension," *Cognition*, vol. 74, no. 1, pp. 33–70, 2000.

[23] S. H. Johnson-Frey, "The neural bases of complex tool use in humans," *Trends in cognitive sciences*, vol. 8, no. 2, pp. 71–78, 2004.

[24] D. Stout and T. Chaminade, "Making tools and making sense: complex, intentional behaviour in human evolution," *Cambridge Archaeological Journal*, vol. 19, no. 01, pp. 85–96, 2009.

[25] A. Cangelosi, G. Metta, G. Sagerer, S. Nolfi, C. Nehaniv, K. Fischer, J. Tani, T. Belpaeme, G. Sandini, F. Nori *et al.*, "Integration of action and language knowledge: A roadmap for developmental robotics," *Autonomous Mental Development, IEEE Transactions on*, vol. 2, no. 3, pp. 167–195, 2010.

[26] M. A. Arbib, "From mirror neurons to complex imitation in the evolution of language and tool use*," *Annual Review of Anthropology*, vol. 40, pp. 257–273, 2011.

[27] J. Steele, P. F. Ferrari, and L. Fogassi, "From action to language: comparative perspectives on primate tool use, gesture and the evolution of human language," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 367, no. 1585, pp. 4–9, 2012.

[28] L. Rat-Fischer, J. K. ORegan, and J. Fagard, "The emergence of tool use during the second year of life," *Journal of experimental child psychology*, vol. 113, no. 3, pp. 440–446, 2012.

[29] P. Willatts, "Development of means–end behavior in young infants: Pulling a support to retrieve a distant object." *Developmental psychology*, vol. 35, no. 3, p. 651, 1999.

[30] D. Stout and T. Chaminade, "The evolutionary neuroscience of tool making," *Neuropsychologia*, vol. 45, no. 5, pp. 1091–1100, 2007.

[31] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 3060–3065.

[32] D. Bullock, S. Grossberg, and F. H. Guenther, "A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm," *Journal of Cognitive Neuroscience*, vol. 5, no. 4, pp. 408–435, 1993.

[33] R. Braud, A. Pitti, and P. Gaussier, "Comparison of absolute and relative strategies to encode sensorimotor transformations in tool-use," in *Development and Learning and on Epigenetic Robotics (ICDL), 2015 IEEE International Conference on*. IEEE, 2015, pp. 267–268.

[34] P. Gaussier and S. Zrehen, "Perac: A neural architecture to control artificial animals," *Robotics and Autonomous Systems*, vol. 16, no. 2, pp. 291–320, 1995.

[35] A. de Rengervé, P. Andry, and P. Gaussier, "Online learning and control of attraction basins for the development of sensorimotor control strategies," *Biological cybernetics*, vol. 109, no. 2, pp. 255–274, 2015.

[36] J. Baillieul, J. Hollerbach, and R. Brockett, "Programming and control of kinematically redundant manipulators," in *Decision and Control, 1984. The 23rd IEEE Conference on*, Dec 1984, pp. 768–774.

[37] F. A. Mussa-Ivaldi and N. Hogan, "Integrable solutions of kinematic redundancy via impedance control," *The International Journal of Robotics Research*, vol. 10, no. 5, pp. 481–491, 1991.

[38] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2001, pp. 298–303.

[39] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural networks*, vol. 11, no. 7, pp. 1317–1329, 1998.

[40] D. M. Wolpert, K. Doya, and M. Kawato, "A unifying computational framework for motor control and social interaction," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 358, no. 1431, pp. 593–602, 2003.

[41] M. Rolf and M. Asada, "What are goals? and if so, how many?" in *Development and Learning and on Epigenetic Robotics, 2015. ICDL 2015. IEEE 5th International Conference on*. IEEE, 2015.

[42] B. Hommel, J. Müsseler, G. Aschersleben, and W. Prinz, "The Theory of Event Coding (TEC): a framework for perception and action planning." *The Behavioral and brain sciences*, vol. 24, no. 5, pp. 849–937, oct 2001.

[43] K. S. Lashley, *The problem of serial order in behavior*. Bobbs-Merrill, 1951.

[44] G. A. Miller, E. Galanter, and K. H. Pribram, "Plans and the structure of behavior," 1960.

[45] R. Cooper and T. Shallice, "Contention scheduling and the control of routine activities," *Cognitive neuropsychology*, vol. 17, no. 4, pp. 297–338, 2000.

[46] M. M. Botvinick, "Hierarchical models of behavior and prefrontal function," *Trends in cognitive sciences*, vol. 12, no. 5, pp. 201–208, 2008.

[47] S. Dehaene and J.-P. Changeux, "A hierarchical neuronal network for planning behavior," *Proceedings of the National Academy of Sciences*, vol. 94, no. 24, pp. 13 293–13 298, 1997.

[48] S. Grossberg, "The adaptive self-organization of serial order in behavior: Speech, language," *Pattern recognition by humans and machines: Speech perception*, vol. 1, p. 187, 2013.

[49] M. N. Nicolescu and M. J. Matarić, "A hierarchical architecture for behavior-based robots," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. ACM, 2002, pp. 227–233.

[50] M. Ogino, M. Hikita, S. Fuke, and M. Asada, "Generation of condition-dependent reaching movements based on layered associative networks," in *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–6.

[51] Y. Sandamirskaya and G. Schöner, "Serial order in an acting system: a multidimensional dynamic neural fields implementation," in *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*. IEEE, 2010, pp. 251–256.

[52] E. A. Billing and Y. Sandamirskaya, "Simultaneous planning and action: Neural-dynamic sequencing of elementary behaviors in robot navigation," *Adaptive Behavior*, 2015.

[53] L. Bonini, F. U. Serventi, L. Simone, S. Rozzi, P. F. Ferrari, and L. Fogassi, "Grasping neurons of monkey parietal and premotor cortices encode action goals at distinct levels of abstraction during complex action sequences," *The Journal of Neuroscience*, vol. 31, no. 15, pp. 5876–5886, 2011.

[54] G. Rizzolatti and L. Craighero, "The mirror-neuron system," *Annu. Rev. Neurosci.*, vol. 27, pp. 169–192, 2004.

[55] L. Craighero, G. Metta, G. Sandini, and L. Fadiga, "The mirror-neurons system: data and models," *Progress in brain research*, vol. 164, pp. 39–59, 2007.

[56] L. Fogassi, P. F. Ferrari, B. Gesierich, S. Rozzi, F. Chersi, and G. Rizzolatti, "Parietal lobe: from action organization to intention understanding," *Science*, vol. 308, no. 5722, pp. 662–667, 2005.

[57] J. Gibson, *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, Inc, Mahwah, NJ, 1979.

[58] S. H. Creem-Regehr and J. N. Lee, "Neural representations of graspable objects: are tools special?" *Cognitive Brain Research*, vol. 22, no. 3, pp. 457–469, 2005.

[59] D. N. Bub, M. E. Masson, and G. S. Cree, "Evocation of functional and volumetric gestural knowledge by objects and words," *Cognition*, vol. 106, no. 1, pp. 27–58, 2008.

[60] R. Ellis and M. Tucker, "Micro-affordance: The potentiation of components of action by seen objects," *British journal of psychology*, vol. 91, no. 4, pp. 451–471, 2000.

[61] P. Cisek, "Cortical mechanisms of action selection: the affordance competition hypothesis," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 362, no. 1485, pp. 1585–1599, 2007.

[62] P. Cisek and J. F. Kalaska, "Neural mechanisms for interacting with a world full of action choices," *Annual review of neuroscience*, vol. 33, pp. 269–298, 2010.

[63] S. Thill, D. Caligiore, A. M. Borghi, T. Ziemke, and G. Baldassarre, "Theories and computational models of affordance and mirror systems: an integrative review," *Neuroscience & BioBehavioral Reviews*, vol. 37, no. 3, pp. 491–521, 2013.

[64] R. Möller and W. Schenck, "Bootstrapping cognition from behaviora computerized thought experiment," *Cognitive Science*, vol. 32, no. 3, pp. 504–542, 2008.

[65] L. W. Barsalou, "Simulation, situated conceptualization, and prediction," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1521, pp. 1281–1289, 2009.

[66] L. W. Barsalou, C. Breazeal, and L. B. Smith, "Cognition as coordinated non-cognition," *Cognitive Processing*, vol. 8, no. 2, pp. 79–91, 2007.

[67] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends in cognitive sciences*, vol. 2, no. 9, pp. 338–347, 1998.

[68] M. Kawato, "Internal models for motor control and trajectory planning," *Current opinion in neurobiology*, vol. 9, no. 6, pp. 718–727, 1999.

[69] D. M. Wolpert and Z. Ghahramani, "Computational principles of movement neuroscience," *nature neuroscience*, vol. 3, pp. 1212–1217, 2000.

[70] M. V. Butz, O. Sigaud, and P. Gerard, "Internal models and anticipations in adaptive learning systems," in *Anticipatory behavior in adaptive learning systems*. Springer, 2003, pp. 86–109.

[71] L. W. Barsalou, "Perceptual symbol systems," *Behavioral and brain sciences*, vol. 22, no. 04, pp. 577–660, 1999.

[72] G. Hesslow, "Conscious thought as simulation of behaviour and perception," *Trends in cognitive sciences*, vol. 6, no. 6, pp. 242–247, 2002.

[73] R. Grush, "The emulation theory of representation: motor control, imagery, and perception," *Behavioral and brain sciences*, vol. 27, no. 03, pp. 377–396, 2004.

[74] J. Decety and J. Grèzes, "The power of simulation: imagining one's own and other's behavior," *Brain research*, vol. 1079, no. 1, pp. 4–14, 2006.

[75] G. Pezzulo, M. Candidi, H. Dindo, and L. Barca, "Action simulation in the human brain: twelve questions," *New Ideas in Psychology*, vol. 31, no. 3, pp. 270–290, 2013.

[76] J. K. O'Regan and A. Noë, "A sensorimotor account of vision and visual consciousness," *Behavioral and brain sciences*, vol. 24, no. 05, pp. 939–973, 2001.

[77] S. Mahé, R. Braud, P. Gaussier, M. Quoy, and A. Pitti, "Exploiting the gain-modulation mechanism in parieto-motor neurons: Application to visuomotor transformations and embodied simulation," *Neural Networks*, vol. 62, pp. 102–111, 2015.

[78] N. Srinivasa and S. Grossberg, "A head–neck–eye system that learns fault-tolerant saccades to 3-d targets using a self-organizing neural model," *Neural Networks*, vol. 21, no. 9, pp. 1380–1391, 2008.

[79] N. Srinivasa, R. Bhattacharyya, R. Sundareswara, C. Lee, and S. Grossberg, "A bio-inspired kinematic controller for obstacle avoidance during reaching tasks with real robots," *Neural Networks*, vol. 35, pp. 54–69, 2012.

[80] L. B. Smith and E. Thelen, "Development as a dynamic system," *Trends in cognitive sciences*, vol. 7, no. 8, pp. 343–348, 2003.

[81] J. R. Flanagan, P. Vetter, R. S. Johansson, and D. M. Wolpert, "Prediction precedes control in motor learning," *Current Biology*, vol. 13, no. 2, pp. 146–150, 2003.

[82] R. Braud, G. Mostafaoui, A. Karaouzene, and P. Gaussier, "Simulating the emergence of early physical and social interactions: A developmental route through low level visuomotor learning," in *From Animals to Animats 13*. Springer, 2014, pp. 154–165.

[83] M. Rolf, J. J. Steil, and M. Gienger, "Learning flexible full body kinematics for humanoid tool use," in *Emerging Security Technologies (EST), 2010 International Conference on*. IEEE, 2010, pp. 171–176.

[84] L. Jamone, B. Damas, N. Endo, J. Santos-Victor, and A. Takanishi, "Incremental development of multiple tool models for robotic reaching through autonomous exploration," *Paladyn, Journal of Behavioral Robotics*, vol. 3, no. 3, pp. 113–127, 2012.

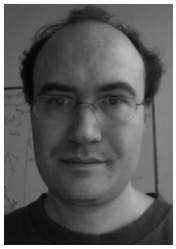
[85] C. Nabeshima, Y. Kuniyoshi, and M. Lungarella, "Adaptive body schema for robotic tool-use," *Advanced Robotics*, vol. 20, no. 10, pp. 1105–1126, 2006.

[86] E. Billing, R. Lowe, and Y. Sandamirskaya, "Simultaneous planning and action: Neural-dynamic sequencing of elementary behaviours in robot navigation," *Adaptive Behavior*, vol. 9, pp. 1–22, 2015.

[87] G. Schöner, "Dynamical systems approaches to cognition," *Cambridge handbook of computational cognitive modeling*, pp. 101–126, 2008.
[88] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 49–73, 2013.
[89] S. Hart and R. Grupen, "Learning generalizable control programs," *Autonomous Mental Development, IEEE Transactions on*, vol. 3, no. 3, pp. 216–231, 2011.
[90] J. Law, P. Shaw, K. Earland, M. Sheldon, and M. Lee, "A psychology based approach for longitudinal development in cognitive robotics," *Frontiers in neurorobotics*, vol. 8, 2014.
[91] F. S. Perotto, "A computational constructivist model as an anticipatory learning mechanism for coupled agent–environment systems," *Constructivist Foundations*, vol. 9, no. 1, pp. 46–56, 2013.
[92] E. Ugur, E. Oztop, and E. Sahin, "Goal emulation and planning in perceptual space using learned affordances," *Robotics and Autonomous Systems*, vol. 59, no. 7, pp. 580–595, 2011.
[93] H. H. Chaput, *The constructivist learning architecture: A model of cognitive development for robust autonomous robots*. Computer Science Department, University of Texas at Austin, 2004.
[94] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen *et al.*, "Object–action complexes: Grounded abstractions of sensory–motor processes," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 740–757, 2011.

Raphael Braud holds an MSc in Intelligent Systems and Robotics from the University of Cergy-Pontoise, France, and he is currently pursuing a PhD degree in Developmental Robotics on the modeling of cognitive mechanisms and sensorimotor approaches for both motor control and tool-use in infants at the ETIS Laboratory, in the same institution. He is now at Aberystwyth University working on the project MoDeL, conducting research on developmental learning for humanoid robots through interactions with objects and tools.

Alexandre Pitti is Maitre de Conferences (associate professor) at the ETIS lab in the Neurocybernetic team of Cergy-Pontoise University, France. He did his PhD at the University of Tokyo (prof Kuniyoshi) and worked as JST researcher in the ERATO project of prof Minoru Asada. His topics are in embodied AI, Developmental and Neuro-Robotics, neural architecture for cognitive systems, multimodal integration, and body representation. He holds the CNRS-UCP chaire of excellence on Cognitive Robotics.



Philippe Gaussier received a PhD in computer science from the University of Paris XI (Orsay) for work on the modeling and simulation of a visual system inspired by mammalian vision. From 1992 to 1994, he conducted research in neural network (NN) applications and in the control of autonomous mobile robots at the Swiss Federal Institute of Technology. He has edited a special issue of the journal Robotics and Autonomous Systems on "moving the frontier between robotics and biology". He is now professor at the Cergy-Pontoise University in France and leads the neurocybernetic team of the image and signal processing laboratory (ETIS). His research interests are focused on the modeling of the cognitive mechanisms involved in visual perception, motivated navigation and action selection. He is also interested in the study of the dynamical interactions between individuals with a particular research in the fields of imitation and emotions.