Intrinsic Motivation for Autonomous Mental Development

Pierre-Yves Oudeyer, Frédéric Kaplan, Véréna Hafner

Intrinsic Motivation Systems for Autonomous Mental Development

Pierre-Yves Oudeyer, Frédéric Kaplan, Verena V. Hafner

Sony Computer Science Lab, Paris

6 rue Amyot 75005 Paris

{oudeyer, kaplan, hafner}@csl.sony.fr

http://www.csl.sony.fr

Abstract

Exploratory activities seem to be intrinsically rewarding for children and crucial for their cognitive development. Can a machine be endowed with such an intrinsic motivation system? This is the question we study in this paper, presenting a number of computational systems that try to capture this drive towards novel or curious situations. After discussing related research coming from developmental psychology, neuroscience, developmental robotics and active learning, this article presents the mechanism of Intelligent Adaptive Curiosity, an intrinsic motivation system which pushes a robot towards situations in which it maximizes its learning progress. This drive makes the robot focus on situations which are neither

too predictable nor too unpredictable thus permitting autonomous mental development. The complexity of the robot's activities autonomously increases and complex developmental sequences self-organize without being constructed in a supervised manner. Two experiments are presented illustrating the stage-like organization emerging with this mechanism. In one of them, a physical robot is placed on a baby play mat with objects that it can learn to manipulate. Experimental results show that the robot first spends time in situations which are easy to learn, then shifts its attention progressively to situations of increasing difficulty, avoiding situations in which nothing can be learnt. Finally, these various results are discussed in relation to more complex forms of behavioural organization and data coming from developmental psychology.

Keywords: intrinsic motivation, curiosity, values, development, learning, autonomy, epigenetic robotics, behaviour, developmental trajectory, complexity, active learning.

1 The challenge of autonomous mental development

All humans develop in an autonomous open-ended manner through life-long learning. So far, no robot has this capacity. Building such a robot is one of the greatest challenges to robotics today, and is the long-term goal of the growing field of developmental robotics ([1, 2]). This article explores a possible route towards such a goal. Our approach is inspired by developmental psychology and

our ambition is to build systems featuring some of the fundamental aspects of an infant's development. More precisely two remarkable properties of human infant development inspire us.

1.1 Development is progressive and incremental

First of all, development involves the progressive increase of the complexity of the activities of children with an associated increase of their capabilities. Moreover, infants' activities always have a complexity which is well-fitted to their current capabilities. Children undergo a developmental sequence during which each new skill is acquired only when associated cognitive and morphological structures are ready. For example, children first learn to roll over, then to crawl and sit, and only when these skills are operational, they begin to learn how to stand. Development is progressive and incremental. Taking inspiration from these observations, some roboticists argue that learning a given task could be made much easier for a robot if it followed a developmental sequence (e.g. "Learning form easy mission" ([3]). But very often, the developmental sequence is crafted by hand: roboticists manually build simpler versions of a complex task and put the robot successively in versions of the task of increasing complexity. For example, if they want to teach a robot the grammar of a language, they first give it examples of very simple sentences with few words, and progressively they add new types of grammatical constructions and complications such as nested subordinates ([4]). This technique is useful in many cases, but has shortcomings which limit our capacity to build robots that develop in an open-ended manner.

Indeed, this is not practical: for each task that one wants the robot to learn, one has to design versions of this task of increasing complexity, and one also has to design manually a reward function dedicated to this particular task. This might be all right if one is interested in only one or two tasks, but a robot capable of life-long learning should eventually be able to perform thousands of tasks. And even if one would engage in such a daunting task of designing manually thousands of specific reward functions, there is another limit. The robot is equipped with a learning machine whose learning biases are often not intuitive: this means that it is also conceptually difficult most of the time to think of simpler versions of a task that might help the robot. It is often the case that a task that one considers to be easier for a robot might turn out in fact to be more difficult.

1.2 Development is autonomous and active

This leads us to a second property of child development from which we should be inspired: it is autonomous and active. Of course, adults help by scaffolding their environment, but this is just a help: eventually, infants decide by themselves what they do, what they are interested in, and what their learning situations are. They are not forced to learn the tasks suggested by adults, they can invent their own. Thus, they construct by themselves their developmental sequence. Anyone who has ever played with an infant in its first year knows for example that it is extremely difficult to get the child to play with a toy that is chosen by the adult if other toys and objects are around. In fact, most often the toys that

4

we think are adapted to them and will please them are not the ones they prefer: they can have much more fun and instructive play experiences with adult objects, such as magazines, keys, or flowers. Also, most of the time, infants engage in particular activities for their own sake, rather than as steps towards solving practical problems. This is indeed the essence of play. This suggests the existence of a kind of intrinsic motivation system, as proposed by psychologists like White ([5]), which provide internal rewards during these play experiences. Such internal rewards are obviously useful, since they are incentives to learn many skills that will potentially be readily available later on for challenges and tasks which are not yet foreseeable.

In order to develop in an open-ended manner, robots should certainly be equipped with capacities for autonomous and active development, and in particular with intrinsic motivation systems, forming the core of a system for task-independent learning. Yet, this crucial issue is still largely underinvestigated. The rest of the article is organized in the following way. The next section presents a general discussion of research related to intrinsic motivation in the domain of psychology, neuroscience, developmental robotics and active learning. Section III presents a critical review and a classification of existing intrinsic motivation systems and determines key characteristics important to permit autonomous mental development. Section IV describes in detail the algorithm of Intelligent Adaptive Curiosity. Section V discusses methodological issues for characterizing the behaviour and performances of such systems. Section

VI presents a first experiment using Intelligent Adaptive Curiosity with a simple simulated robot. Section VII presents a second more complex experiment involving a physical robot discovering affordances about entities in its environment. Section VIII discusses the results obtained in these two experiments in relation to more complex issues associated with behavioural organization and observation in infant development.

2 Background

2.1 Psychology

White ([5]) presents an argumentation explaining why basic forms of motivation such as those related to the need for food, sex or physical integrity maintenance cannot account for an animal's exploratory behaviour, and in particular for humans. He proposed rather that exploratory behaviours can be by themselves a source of rewards. Some experiments have been conducted showing that exploration for its own sake is an activity which is not always a secondary reinforcer: it is certainly a built-in primary reinforcer. The literature on education and development also stresses the distinction between intrinsic and extrinsic motivations ([6]). Psychologists have proposed possible mechanisms which explain the kind of exploratory behaviour that for example humans show. Berlyne ([7]) proposed that exploration might be triggered and rewarded for situations which include novelty, surprise, incongruity and complexity. He also refined this idea by observing that the most rewarding situations were those with an interme-

diate level of novelty, between already familiar and completely new situations. This theory has strong resonance points with the theory of flow developed by Csikszentmihalyi ([8]) which argues that a crucial source of internal rewards for humans is the self-engagement in activities which require skills just above their current level. Thus, for Csikszentmihalyi, exploratory behaviour can be explained by an intrinsic motivation for reaching situations which represent a learning challenge. Internal rewards are provided when a situation which was previously not mastered becomes mastered within an amount of time and effort which must be not too small but also not too large. Indeed, in analogy with Berlyne ([7]), Csikszentmihalyi insists that the internal reward is maximal when the challenge is not too easy but also not too difficult.

2.2 Neuroscience

Recent discoveries showing a convergence between patterns of the activity in the midbrain dopamine neurons and computational model of reinforcement learning have led to an important amount of speculations about learning activities in the brain ([9]). Central to some of these models is the idea that dopamine cells report the error in predicting expected reward delivery. Most experiments in this domain focus on the involvement of dopamine for predicting extrinsic (or external) reward (e.g. food). Yet, recently some researchers provided ground for the idea that dopamine might also be involved in the processing of types of intrinsic motivation associated with novelty and exploration ([10], [11]). In particular, some studies suggest that dopamine responses could be interpreted as report-

ing "prediction error" (and not only "reward prediction error") ([12]). These findings supports the idea that intrinsic motivation systems could be present in the brain in some forms or another and that signals reporting prediction error could play a critical role in this context.

2.3 Developmental robotics

Given this background, a way to implement an intrinsic motivation system might be to build a mechanism which can evaluate operationally the degree of "novelty", "surprise", "complexity" or "challenge" that different situations provide from the point of view of a learning robot, and then measuring an associated reward ideally being maximal when these features are in an intermediate level, as proposed by Berlyne ([7]) and Csikszentmihalyi ([13]). Autonomous and active exploratory behaviour can then be achieved by acting so as to reach situations which maximize this measure. The difficult task becomes to find a sensible manner to operationalize the concepts behind the words "novelty", "complexity", "surprise" or "challenge" which are only verbally described and often vaguely defined in the psychology literature.

Only a few researchers have suggested such implementations, and even fewer have tested them on real robots. Typically, they call these systems of autonomous and active exploratory behaviour "artificial curiosity". Schmidhuber, Thrun and Hermann ([14], [15], and [16]) provided initial implementations of artificial curiosity, but they did not integrate this concept within the problematic of developmental robotics, in the sense that they were not concerned with

8

the emergent development sequence and with the increase of the complexity of their machines (and they did not use robots, but learning machines on some abstract problems). They were only concerned in how far artificial curiosity can speed up the acquisition of knowledge. The first integrated view of developmental robotics that incorporated a proposal for a novelty drive was described by Weng and colleagues ([17]; [18]). Then, Kaplan and Oudeyer proposed an implementation of artificial curiosity within a developmental framework ([19]), and Marshall, Blank and Meeden as well as Barto, Sing and Chentanez suggested variations on the novelty drive ([20], [21]). As we will explain later on in the paper, these pioneering systems have a number of limitations making them impossible to use on real robots in real uncontrolled environments. Furthermore, to our knowledge, it has not yet been shown how they could successfully lead to the autonomous formation of a developmental sequence comprising more than one stage. This means that typically they have allowed for the development and emergence of one level of behavioural patterns, but did not show how new levels of more complex behavioural patterns could emerge without the intervention of a human or a change in the environment provoked by a human.

2.4   Active Learning

Interestingly, the mechanisms developed in these papers devoted to the implementation of artificial curiosity have strong similarities with mechanisms developed in the field of statistics, where it is called "optimal experiment design" ([22]), and in machine learning, where it is called "active learning" ([23], [24]).

In these contexts, the problem is summarized with the question: how to choose the next example for a learning machine in order to to minimize the number of examples necessary to achieve a given level of performance in generalization? Or said another way: how to choose the next example so that the gain in information for the machine learner will be maximal? A number of techniques developed in active learning have proved to speed up significantly the learning of machines (e.g. [25], [26], [27], [28], [29], [30], [31]) and even to allow performance on generalization which are not possible with passive learning ([32]). Yet, these techniques were developed for applications in which the mapping to be learnt was clean and typically presented as pre-processed well-prepared datasets. They are also typically based on mathematical theory like Optimal Experiment Design which assumes that the noise is independently normally distributed ([33]). On the contrary, the domain that real robots shall investigate is the real unconstrained world, which is a highly complicated and "muddy" structure, as pointed out by Weng ([34]), full of very different kinds of intertwined non-gaussian inhomogeneous noise. As a consequence, these methods cannot be used directly in the developmental robotics domain, and there is no obvious way to extend them in this direction. Moreover, there exists no efficient implementation for methods like optimal experiment design in continuous spaces, and already in discrete spaces the computational cost is high ([35]).

3 Existing intrinsic motivation systems

Existing computational approaches to intrinsic motivations and artificial curiosity are typically based on an architecture which comprises a machine which learns to anticipate the consequences of the robot's actions, and in which these actions are actively chosen according to some internal measures related to the novelty or predictability of the anticipated situation. Thus, the robots in these approaches can be described as having two modules: 1) one module implements a learning machine **M** which learns to predict the sensorimotor consequences when a given action is executed in a given sensorimotor context; 2) another module is a meta learning machine **metaM** which learns to predict the errors that machine **M** makes in its predictions: these meta-predictions are then used as the basis of a measure of the potential interestingness of a given situation. The existing approaches can be divided into three groups, according to the way action-selection is made depending on the predictions of **M** and **metaM**.

3.1 Group 1: Error maximization

In the first group (e.g. [18]; [15]; [20], [21]) robots directly use the error predicted by **metaM** to choose which action to do[1]. The action that they choose at each step is the one for which **metaM** predicts the largest error in prediction of M. This has shown to be efficient when the machine **M** has to learn a mapping which is learnable, deterministic and with homogeneous Gaussian noise ([32];

---

[1]Of course, we are only talking about the "novelty" drive here: their robots are sometimes equipped with other competing drives or can respond to external human based reward sources.

[15]; [17]; [21]). But this method shows limitations when used in a real uncontrolled environment. Indeed, in such a case, the mapping that $\mathbf{M}$ has to learn is not anymore deterministic, and the noise is vastly inhomogeneous. Practically, this means that a robot using this method will for example be stuck by white noise or more generally by situations which are inherently too complex for its learning machinery or situations for which the causal variables are not perceivable or observable by the robot. For example, a robot equipped with a drive which pushes it towards situations which are maximally unpredictable might discover and stay focused on movement sequences like running fast against a wall, the shock resulting in an unpredictable bounce (in principle, the bounce is predictable since it obeys the deterministic laws of classic mechanics but in practice this prediction requires the perfect knowledge of all the physical properties of the robot body as well as those of the wall, which is typically far from being the case for a robot). So, in uncontrolled environments, a robot equipped with this intrinsic motivation system will get stuck and display behaviours which do not lead to development and that can sometimes even be dangerous.

3.2 Group 2: Progress maximization

A second group of models tried to avoid getting stuck in the presence of pure noise or unlearnable situations by using more indirectly the prediction of the error of $\mathbf{M}$ (e.g. [16, 19]). In these models a third module that we call $\mathbf{KGA}$ for Knowledge Gain Assessor is added to the architecture. Figure 1 shows an illustration of these systems. This new module enhances the capabilities of the

meta-machine **metaM**: **KGA** predicts the mean error rate of **M** in the close future and in the next sensorimotor contexts. **KGA** also stores the recent mean error rate of **M** in the most recent sensorimotor contexts. The crucial point of these models is that the interestingness of candidate situations are evaluated using the difference between the expected mean error rate of the predictions of **M** in the close future, and the mean error rate in the close past. For each situation that the robot encounters, it is given an internal reward which is equal to the inverse of this difference (which also corresponds to the local derivative of the error rate curve of **M**). This internal reward is positive when the error rate decreases, and negative when it increases. The motivation system of the robot is then a system in which the action chosen is that for which **KGA** predicts that it will lead to the greatest decrease of the mean error rate of **M**. This ensures that the robot will not stay in front of white noise for a long time or in unlearnable situations because this does not lead to a decrease of its errors in prediction.

However, this method has only been tested in spaces in which the robot can do only one kind of activity, such as for example moving the head and learning to predict the position of high luminance points ([19]). But the ideal characteristic of a developmental robot is that it may engage in various kinds of activities, such as learning to walk, learning to grip things in its hand, learning to track a visual target, learning to catch the attention of other social beings, learning to vocalize, etc. In such cases, the robot can typically switch rapidly from one activity to the other: for example, making a trial at gripping an object

13

that it sees and suddenly shifting to trying to track the movement of another being in its environment. In such a case, measuring the evolution in time of its performance in predicting what happens will lead to a measure which is hardly interpretable. Indeed, using the method we described in the last paragraph will make the robot compare its error rate in anticipation while it is trying to grip an object with its error rate in anticipation while it is trying to anticipate the reaction of the other being when he vocalizes, if these two kinds of activities are sequenced. Thus, it will often lead the robot to compare its performances for activities which are of a different kind, which has no obvious meaning. And indeed, using this direct measure of the decrease in the error rate in prediction will provide the robot with internal rewards when shifting from an activity with a high mean error rate to activities with a lower mean error rate, which can be higher than the rewards corresponding to an effective increase of the skills of the robot in one of the activities. This will push the robot towards instable behaviour, in which it focuses on the sudden shifts between different kinds of activities rather than concentrate on the actual activities.

3.3   Group 3: Similarity-based progress maximization

Changes are needed so that methods based on the decrease of the error rate in prediction can still work in a realistic complex developmental robotics set-up. It is necessary that the robot monitors the evolution of its error rate in prediction in situations which are of the same kind. It will not anymore compare its current error rate with its error rate in the close past, whatever the current sit-

uation and the situation in the close past are. The similarity between situations must be taken into account. Building a system which can do that correctly represents a big challenge. Indeed, a developmental robot will not be given an innate mechanism with a pre-programmed set of kinds of situations and a mechanism for categorizing each particular situation into one of these kinds. A developmental robot has to be able to build by itself a measure of the similarity of situations and ultimately an organization of the infinite continuous space of particular situations into higher-level categories (or kinds) of situations. For example, a developmental robot does not know initially that on the one hand there can be the "gripping objects" kind of activity and on the other hand the "vocalizing to others" kind of activity. Initially, the world is just a continuous stream of sensations and low-level motor commands for the robot.

A related approach, but with an active learning point of view rather than a developmental robotics point of view, was proposed presenting an implementation of the idea of evaluation the learning progress by monitoring the evolution of the error rate in similar situations ([14]). The implementation described was tested for discrete environments like a two-dimensional grid virtual world on which an agent could move and do one of four discrete actions. The similarity of two situations was evaluated by a binary function stating whether they correspond exactly to the same discrete state or not. From an active learning point of view, it was shown that in this case the system can significantly speed up the learning, even if some parts of the space are pure noise. This system was not studied under the developmental robotics point of view: it was not

shown whether this allowed for a self-organization of the behaviour of the robot into a developmental sequence featuring clearly several stages of increasing complexity. Moreover, because the system was only tested on a discrete simulated environment, it is difficult to generalize the results to the general case in which the environment and action spaces are continuous, and where two situations are never numerically exactly the same. Nevertheless, this article suggests a possible manner to use this method in continuous spaces. It is based on the use of a learning machine such as a feed-forward neural network which takes as input a particular situation and predicts the error associated with the anticipation of the consequence of a given action in this situation. This measure is then used in a formula to evaluate the learning progress. Thanks to the generalization properties of a machine like a neural network, the author claims that the mechanism will correctly generalize the evaluation of learning progress from one situation to similar situations. Yet, it is not clear how this will work in practice since the error function, and thus the learning progress function, is locally highly non-stationary. This provokes a risk of over-generalization. Another limit of this work resides within the particular formula that is used to evaluate the learning progress associated with a candidate situation, which consists in making the difference between the error in the anticipation of this situation before it has been experienced and the error in the anticipation of exactly the same situation after it has been experienced. On the one hand, this can only work for a learning machine with a low learning rate, as pointed out by the author, and will not work with for example one-shot learning of memory-based methods. On the

other hand, considering the state of the learning machine just before and just after one single experience can possibly be sensitive to stochastic fluctuations.

The next section will present a system which provides an implementation of the idea of evaluating the learning progress by comparing similar situations. This system is made to work in continuous spaces, and we will actually show that this system works both in a virtual robot set-up and in a real robotic set-up with continuous motor and/or perceptual spaces. One of its crucial features is that it introduces a mechanism of situation categorization, which splits the space incrementally and autonomously into different regions which correspond to different kinds of activities from the point of view of the robot. This allows to compare the similarity of two situations not directly based on their intrinsic metric distance, but on their belonging to a given situation category. Another feature is the fact that we monitor in each of these regions the evolution of the error rate in prediction for an extended period of time, which allows to use smoothing procedures and avoid problems due to stochastic fluctuations. The "regional" evaluation of similarity combined with the smoothing of the error rate curve is a way to cope with the non-stationarity of the learning progress function. Another feature is that it makes no pre-supposition on the learning rate of the learning machines, and thus can be used with one-shot learning methods like nearest neighbours algorithms as well as with slowly learning neural networks for example.

4   Intelligent Adaptive Curiosity

The system described in this section is called **Intelligent Adaptive Curiosity** (IAC):

- it is a motivation, or **drive**, in the same sense than food level maintenance or heat maintenance are drives, but instead of being about the maintenance of a physical variable, the IAC drive is about the maintenance of an abstract dynamic cognitive variable: **the learning progress**, which must be kept maximal. This definition makes it an intrinsic motivation.

- it is called **curiosity** because maximizing the learning progress pushes (as a side effect) the robot towards novel situations in which things can be learnt.

- it is **adaptive** because the situations that are attractive change over time: indeed, once something is learnt, it will not provide learning progress anymore.

- it is called **intelligent** because it keeps, as a side effect, the robot away both from situations which are too predictable and from situations which are too unpredictable (i.e. the edge of order and chaos in the cognitive dynamics). Indeed, thanks to the fact that one evaluates the learning progress by comparing situations which are similar and in a "regional" manner, the pathologic behaviours that we described in the previous section are avoided.

We will now describe how this system can be fully implemented. This implementation can be varied in many manners, for example by replacing the implementation of the learning machines **M**, **metaM** and **KGA**. The one we provide is basic and was developed for its practical efficiency. Also, it will be clear to the reader that in an efficient implementation, the machines **M**, **metaM** and **KGA** are not easily separable (keeping them separate entities in the previous paragraphs was for reasons of keeping the explanation easier to understand).

4.1 Summary

IAC relies on a memory which stores all the experiences encountered by the robot in the form of vector exemplars. There is a mechanism which incrementally splits the sensorimotor space into regions, based on these exemplars. Each region is characterized by its exclusive set of exemplars. Each region is also associated with its own learning machine, which we call an expert. This expert is trained with the exemplars available in its region. When a prediction corresponding to a given situation has to be made by the robot, then the expert of the region which covers this situation is picked up and used for the prediction. Each time an expert makes a prediction associated to an action which is actually executed, its error in prediction is measured and stored in a list which is associated to its region. Each region has its own list. This list is used to evaluate the potential learning progress that can be gained by going in a situation covered by its associated region. This is made based on a smoothing of the list of errors, and on an extrapolation of the derivative. When in a given situation, the robot

19

creates a list of possible actions and chooses the one for which it evaluates it will lead to a situation with maximal expected learning progress[2]

4.2 Sensorimotor apparatus

The robot has a number of real-valued sensors $s_i(t)$ which are here summarized by the vector $\mathbf{S(t)}$. Its actions are controlled by the setting of the real number values of a set of action/motor parameters $m_i(t)$, which we summarize using the vector $\mathbf{M(t)}$. These action parameters can potentially be very low level (for example the speed of motors) or of a higher-level (for example the control parameters of motor primitives such as the biting or bashing movement that we will describe in the section devoted to the "Playground Experiment"). We denote the sensorimotor context $\mathbf{SM(t)}$ as the vector which summarizes the values of all the sensors and the action parameters at time t (it is the concatenation of $\mathbf{S(t)}$ and $\mathbf{M(t)}$). In all that follows, there is an internal clock in the robot which discretized the time, and new actions are chosen at every time step.

---

[2]A variant of this system is the use of only one monolithic learning system, keeping the mechanism of region construction by incremental space splitting. In this case, for each prediction of the single learning system, its error is stored in the list corresponding to the region covering the associated situation. The evaluation of the expected learning progress of a candidate situation is the same as in the system presented here. Yet, we prefer to use one learning system per region in order to avoid forgetting problems which are typical of monolithic learning machines when used in a life-long learning set-up with various kinds of situations.

4.3 Regions

IAC equips the robot with a memory of all the exemplars $(\mathbf{SM(t)}, \mathbf{S(t+1)})$ which have been encountered by the robot. There is a mechanism which incrementally splits the sensorimotor space into regions, based on these exemplars. Each region is characterized by its exclusive set of exemplars. At the beginning, there is only one region $\mathcal{R}_1$. Then, when a criterion C_1 is met, this region is split into two regions. This is done recursively. A very simple criterion C_1 can be used: when the number of exemplars associated to the region is above a threshold $T = 250$, then split. This criterion allows to guarantee a low number of exemplars in each leaf, which renders the prediction and learning mechanism that we will describe in the next paragraphs computationally efficient. The counterpart is that it will lead to systems with many regions which are not easily interpretable from a human point of view.

When a splitting has been decided, then another criterion C_2 must be used to find out how the region will be split. Again, the choice of this criterion was made so that it is computationally and experimentally efficient. The idea is that we split the set of exemplars into two sets so that the sum of the variances of $\mathbf{S(t+1)}$ components of the exemplars of each set, weighted by the number of exemplars of each set, is minimal. Let us explain this mathematically. Let us denote

$$\Gamma_n = \{(\mathbf{SM(t)}, \mathbf{S(t+1)})_i\}$$

the set of exemplars possessed by region $\mathcal{R}_n$. Let us denote j a cutting dimension and v_j an associated cutting value. Then the split of Γ_n into Γ_{n+1} and Γ_{n+2} is

done by choosing a j and a v_j such that (criterion C_2):

- all the exemplars $(\mathbf{SM(t)}, \mathbf{S(t+1)})_i$ of Γ_{n+1} have the jth component of their $\mathbf{SM(t)}$ smaller than v_j;

- all the exemplars $(\mathbf{SM(t)}, \mathbf{S(t+1)})_i$ of Γ_{n+2} have the jth component of their $\mathbf{SM(t)}$ greater than v_j;

- the quantity

  $|\Gamma_{n+1}|.\sigma(\{\mathbf{S(t+1)}|(\mathbf{SM(t)}, \mathbf{S(t+1)}) \in \Gamma_{n+1}\})+$

  $|\Gamma_{n+2}|.\sigma(\{\mathbf{S(t+1)}|(\mathbf{SM(t)}, \mathbf{S(t+1)}) \in \Gamma_{n+2}\})$

  is minimal, where

  $$\sigma(\mathcal{S}) = \frac{\sum_{v \in \mathcal{S}} ||v - \frac{\sum_{v \in \mathcal{S}} v}{|\mathcal{S}|}||^2}{|\mathcal{S}|}$$

  where $\mathcal{S}$ is a set of vectors and $|\mathcal{S}|$ denotes the cardinal of $\mathcal{S}$.

Then recursively and for each region, if the criterion C_1 is met, the region is split into two regions with the criterion C_2. This is illustrated in figure 2.

Each region stores all the cutting dimensions and the cutting values that were used in its generation as well as in the generation of its parent experts. As a consequence when a prediction has to be made of the consequences of $\mathbf{SM(t)}$, it is easy to find out the expert specialist for this case: it is the one for which $\mathbf{SM(t)}$ satisfies all the cutting tests (and there is always a single expert which corresponds to each $\mathbf{SM(t)}$).

4.4 Experts

To each region $\mathcal{R}_n$, there is an associated learning machine $\mathbf{E_n}$, called an expert. A given expert $\mathbf{E_n}$ is responsible for the prediction of $\mathbf{S(t+1)}$ given $\mathbf{SM(t)}$ when $\mathbf{SM(t)}$ is a situation which is covered by its associated region $\mathcal{R}_n$. Each expert $\mathbf{E_n}$ is trained on the set of exemplars which is possessed by its associated region $\mathcal{R}_n$. An expert can be a neural-network, a support-vector machine or a Bayesian machine for example. For all learning machines whose training can be incremental, such as neural networks, support-vector machines, or memory-based methods, then the system is efficient since it is not necessary to re-train each expert on all the exemplars of each region, but just to update one single expert by feeding the new exemplar to it. Still, when a region is split, one cannot use directly the "parent" expert to implement the two children experts. Each child expert is typically a fresh expert re-trained with the exemplars that its associated region has inherited. The computational cost associated with this is limited thanks to the fact that the number of exemplars is never higher than $T = 250$ as guaranteed by the C_1 criterion.[3]

---

[3]Even computationally demanding learning machines such as non-linear support vector machines require only a few dozens milliseconds on a standard computer to be trained with 250 examples, even if these examples have several hundred dimensions ([36]). In the experiments described in the next sections, we use a very simple learning algorithm for implementing the expert: the nearest-neighbours algorithm. In this case, there is not even a need for re-training the expert, since the expert *is* the set of exemplars. In general, the use of the nearest-neighbour algorithm is computationally costly when used at the prediction stage, since it requires as many computations of distances as there are exemplars. Again, the criterion C_1 guarantees that the number of exemplars is always low and allows for a fast computation of

4.5 Evaluation of learning progress

This partition of the sensorimotor space into different regions is the basis of our regional evaluation of learning progress. Each time an action is executed by the robot in a given sensorimotor context $\mathbf{SM(t)}$ covered by the region $\mathcal{R}_n$, the robot can measure the discrepancy between the sensory state $\widetilde{\mathbf{S}}(\mathbf{t+1})$ that the expert $\mathbf{E_n}$ predicted and the actual sensory state $\mathbf{S(t+1)}$ that it measures. This provides a measure of the error of the prediction of $\mathbf{E_n}$ at time $t+1$:

$$e_n(t+1) = ||\mathbf{S(t+1)} - \widetilde{\mathbf{S}}(\mathbf{t+1})||^2$$

the closest exemplar. It is also interesting to note that if one would use a monolithic learning system with only one global expert, which is a variation of IAC mentioned earlier, then the use of the nearest neighbours algorithm would become soon computationally very expensive since a life-long learning robot can accumulate millions of exemplars. On the contrary, using local experts to which access is computed with a tree of cheap numerical comparisons (see figure 2) allows to compute approximately correct global nearest neighbours with a logarithmic complexity ($O(log(N))$) rather than with a linear complexity ($log(N)$). And in fact, using a tree structure with local experts not only allows to speed up the nearest neighbours algorithm, but it also allows to increase the performances in generalization. In practice, this means that the system we present in this paper, when used for example with the nearest neighbours algorithm, can update itself as well as make predictions when it already possesses 3000000 exemplars in a few milliseconds on a personal computer, since in this case it requires about 17 scalar comparisons (depth of the corresponding balanced tree) and 250 distance computation between points. Admittedly, this requires a lot of memory, but it is interesting to note that the collection of 3000000 exemplars composed of for example 20 dimensions, which would take approximately 34 days in the case of the robots presented in the "Playground Experiment" section, would require about $230Mb$ in memory, which is much less than the capacity of most hand held computers nowadays.

This squared error is added to the list of past squared errors of $\mathbf{E_n}$, which are stored in association to the region $\mathcal{R}_n$. We denote this list:

$$e_n(t), e_n(t-1), e_n(t-2), ..., e_n(0)$$

Note that here t denotes a time which is specific to the expert, and not to the robot: this means that $e_n(t-1)$ might correspond to the error made by the expert $\mathbf{E_n}$ in an action performed at $t-10$ for the robot, and that no actions corresponding to this expert were performed by the robot since that time. These lists associated to the regions are then used to evaluate the learning progress that has been achieved after an action $\mathbf{M(t)}$ has been achieved in sensory context $\mathbf{S(t)}$, leading to a sensory context $\mathbf{S(t+1)}$. The learning progress that has been achieved through the transition from the $\mathbf{SM(t)}$ context, covered by region $\mathcal{R}_n$, to the context with a perceptual vector $\mathbf{S(t+1)}$ is computed as the smoothed derivative of the error curve of $\mathbf{E_n}$ corresponding to the acquisition of its recent exemplars. Mathematically, the computation involves two steps:

- the mean error rate in prediction is computed at $t+1$ and $t+1-\tau$:

$$< e_n(t+1) >= \frac{\sum_{i=0}^{\theta} e_n(t+1-i)}{\theta+1}$$

$$< e_n(t+1-\tau) >= \frac{\sum_{i=0}^{\theta} e_n(t+1-\tau-i)}{\theta+1}$$

  where τ is a time window parameter typically equal to 15, and θ a smoothing parameter typically equal to 25.

- the actual decrease in the mean error rate in prediction is defined as:

$$D(t+1) =< e_n(t+1) > - < e_n(t+1-\tau) > \qquad (1)$$

We can then define the actual learning progress as

$$L(t+1) = -D(t+1) \tag{2}$$

Eventually, when a region is split into two regions, both new regions inherit the list of past errors from their parent region, which allows them to make evaluation of learning progress right from the time of their creation.

4.6 Action selection

We have now in place a prediction machinery and a mechanism which provides an internal reward (positive or negative)

$$r(t) = L(t)$$

each time an action is performed in a given context, depending on how much learning progress has been achieved[4]. The goal of the intrinsically motivated robot is then to maximize the amount of internal reward that it gets. Mathematically, this can be formulated as the maximization of future expected rewards (i.e. maximization of the return), that is

$$E\{\sum_{t \geq t_n} \gamma^{t-t_n} r(t))\}$$

---

[4]To integrate reward resulting from learning progress with other kinds of (possibly extrinsic) rewards, a weighted sum can be used. A parameter α_i specifies the relative weight of each reward type.

$$r(t) = \sum_i \alpha_i . r_i(t) \tag{3}$$

.

where γ $(0 \leq \gamma \leq 1)$ is the discount factor, which assigns less weigh on the reward expected in the far future.

This formulation corresponds to a reinforcement learning problem formulation [37] and thus the techniques developed in this field can be used to implement an action selection mechanism which will allow the robot to maximize future expected rewards efficiently. Indeed, in reinforcement learning models, a controller chooses which action a to take in a context s based on rewards provided by a *critic*. Traditional models view the *critic* as being external to the agent. Such situations correspond to extrinsically motivated forms of learning. But the critic can as well be part of the agent itself (as clearly argued by Sutton and Barto [37] p.51-54). As a consequence, the algorithm described in this section can be interpreted as a critic capable of producing internal rewards $r(t)$ in order to guide the agent in its development. Thus, any existing reinforcement learning technique can be associated with the IAC drive.

One simple example would be to use Watkins' Q-learning [38]. The algorithm learns an action-value function $Q(s,a)$, estimating how good it is to perform a given action a ($\mathbf{M(t)}$ in our context) in a given contextual state s ($\mathbf{S(t)}$ in our context). "Good" actions are expected to lead to more future rewards (e.g. more future learning progress in our context). The algorithm can be described in the following procedural form:

- Initialise $Q(s,a)$ with small random uniform values

- Repeat

– In situation s, choose a using a policy derived from Q. For instance choose a that maximize Q in most cases but every once in a while, with a probability ϵ instead select an action at random, uniformly (this is called an ϵ - greedy action selection rule [37])

– Perform action a, observe r and the resulting state s'

– $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \cdot max_{a'}(Q(s',a')) - Q(s,a)]$

– $s \leftarrow s'$

where the parameter α is the learning rate controlling how fast the action-value function is updated by experience. Of course, all the complex issues traditionally encountered in reinforcement learning like trade-off between exploration and exploitation stay crucial for systems using internal rewards based on intrinsic motivation.

The purpose of this article is to focus on the study and understanding of the learning progress definition that we presented. Using a complex reinforcement machinery brings complexity and biases which are specific to a particular method, especially concerning the way they process delayed rewards. While using such a method with intrinsic motivation systems will surely be useful in the future, and is in fact an entire subject of research as illustrated by the work of Barto, Singh and Chentanez ([21]) who have studied the use of sophisticated re-inforcement learning techniques on a simple novelty-based intrinsic motivation system, we will make now a simplification which will allow us not to use such sophisticated re-inforcement learning methods so that the

results we will present in the experiment section can be interpreted more easily. Indeed, this is a necessary step since our intrinsic motivation system involves a non-trivial measure of learning progress which must be carefully understood. This simplification consists in having the system try to maximize only the expected reward it will receive at $t + 1$, i.e. $E\{r(t + 1)\}$ This permits to avoid problems related to delayed rewards and it makes it possible to use a simple prediction system which can predict $r(t + 1)$, and so evaluate $E\{r(t + 1)\}$, and then be used in a straightforward action selection loop. The method we use to evaluate $E\{r(t+1)\}$ given a sensory context $\mathbf{S(t)}$ and a candidate action $\widetilde{\mathbf{M}}(\mathbf{t})$, constituting a candidate sensorimotor context $\widetilde{\mathbf{SM}}(\mathbf{t})$ covered by region $\mathcal{R}_n$, is straightforward but revealed to be efficient: it is equal to the learning progress that was achieved in $\mathcal{R}_n$ with the acquisition of its recent exemplars, i.e.

$$E\{r(t + 1)\} \approx L(t - \theta_{R_n}) \tag{4}$$

where $t - \theta_{R_n}$ is the time corresponding to the last time region $\mathcal{R}_n$ and expert $\mathbf{E_n}$ processed a new exemplar.

Based on this predictive mechanism, one can deduce a straightforward mechanism which manages action selection in order to maximize the expected reward at $t + 1$:

- in a given sensory $\mathbf{S(t)}$ context, the robot makes a list of the possible actions $\widetilde{\mathbf{M}}(\mathbf{t})$ which it can do; If this list is infinite, which is often the case since we work in continuous action spaces, a sample of candidate actions

29

is generated;

- each of these candidate actions $\widetilde{\mathbf{M}}(\mathbf{t})$ associated with the context makes a candidate $\widetilde{\mathbf{SM}}(\mathbf{t})$ vector for which the robot finds out the corresponding region $\mathcal{R}_n$; then the formula we just described is used to evaluate the expected learning progress $E\{r(t+1)\}$ that might be the result of executing the candidate action $\widetilde{\mathbf{M}}(\mathbf{t})$;

- the action for which the system expects the maximal learning progress is chosen and executed except in some cases when a random action is selected (ϵ - greedy action selection rule). In the following experiments ϵ is typically 0.35.

- after the action has been executed and the consequences measured, the system is updated.

5 Methodological issues for measuring behavioural complexity

From a developmental robotics point of view, intrinsic motivation systems are interesting as a way to achieve a continuous increase in behavioural complexity. This raises issues for finding adequate methods to evaluate such systems. Evaluation based on performance level for a set of predefined tasks are the most common way to assess learning progress of adaptive robots. However, as intrinsic motivation systems are designed to result in task-independent autonomous

development, using an evaluation paradigm coming from task-oriented design is not well adapted. Moreover, such evaluation methods are associated with the tempting anthropomorphic bias to evaluate how well robots manage to learn the tasks that humans can learn.

The issue is therefore to evaluate the increase of a robot's behavioural complexity during a developmental sequence. It is important to stress that there is not a single objective way for assessing the increase of complexity of a system. Complexity is always related to a given observer ([39]). Three complementary approaches can be envisioned.

- First, it is possible to evaluate the increase in complexity from the *robot's point of view*. This means measuring internal variables that account for the open-endedness of its development (e.g. cumulative amount of learning progress, evolution of the performance of anticipations, evolution of the way sensorimotor situations are categorized and represented).

- Second, behavioural complexity can be measured from an *external point of view* based on various complexity measures (information-theoretical measures such as the ones presented by Sporns and Pegors could be used in that respect ([40]). The increase in behavioural complexity is assessed by pattern changes in these measures.

- Finally, the experimenter can adopt a method more similar to one used by a psychologist, interpreting developmental sequences as a set of successive *stages*. The stages of development introduced by Piaget are among the

most famous examples of such qualitative descriptions [41]. Each transition between stages corresponds to a broad change in the structure or logic of children's intelligence and/or behaviour. Based on clinical observations, dialogues and small-scale experiments, the psychologist tries to interpret the signs of an internal reorganization. Therefore, the issue is to map external observations to a series of pre-existing interpretative models. Transitions are most of the time progressive and cutting a developmental sequences into sharp division is usually difficult.

The following experiments will illustrate how a combination of some of these methods can be used to assess the development of a robot with an intrinsic motivation system.

6 A first experiment with a simple simulated robot

We present here a robotic simulation implemented with the Webots simulation software ([42]). The purpose of this initial simulated experiment is to show and understand in detail the working of the IAC system in a continuous sensorimotor environment in which there are parts which are clearly inhomogeneous from the learning point of view: there is a part of the space which is easy to learn, a part of the space which contains more complex structures which can be learnt, and a part of the space which is unlearnable.

6.1   Motor control

The robot is a box with two wheels (see figure 3). Each wheel can be controlled by setting its speed (real number between -1 and 1). The robot can also emit a sound of a particular frequency. The action space is 3-dimensional and continuous, and deciding for an action consists in setting the values of the motor vector $\mathbf{M(t)}$:

$$\mathbf{M(t)} = (l,\ r,\ f)$$

where l is the speed of the motor on the left, r the speed of the motor on the right, and f the frequency of the emitted sound. The robot moves in a room. There is a toy in this room that can also move. This toy moves randomly if the sound emitted by the robot has a frequency belonging to zone $f_1 = [0; 0.33]$. It stops moving if the sound is in zone $f_2 = [0, 34; 0, 66]$. The toy jumps into the robot if the sound is in zone $f_3 = [0, 67; 1]$.

6.2   Perception

The robot perceives the distance to the toy with simulated infra-red sensors, so its sensory vector $\mathbf{S(t)}$ is one-dimensional:

$$\mathbf{S(t)} = (d)$$

where d is the distance between the robot and the toy at time t.

6.3   Action perception loop

As a consequence, the mapping that the robot is trying to learn is:

33

$$f \: : \: \mathbf{SM(t)} = (l, \, r, \, f, \, d) \longmapsto \mathbf{S(t+1)} = (\widetilde{d})$$

Using the IAC algorithm, the robot will thus act in order to maximize its learning progress in terms of predicting the next toy distance. The robot has no prior knowledge, and in particular it does not know that there is a qualitative difference between setting the speed of the wheels and setting the sound frequency (for the robot, these are unlabeled motor channels). It does not know that there are three zones of the sensory-motor space of different complexities: the zone corresponding to sounds in f_1, where the distance to the toy cannot be predicted since its movement is random; the zone with sounds in f_3, where the distance to the toy is easy to learn and predict (it is always 0 plus a noise component because Webots simulates the imprecision of sensors and actuators); and the zone with sounds in f_2, where the distance to the toy is predictable (and learnable) but complex and dependant of the setting of the wheel speeds.

Yet, we will now show that the robot manages to autonomously discover these three zones, evaluate their relative complexity, and exploit this information for organizing its own behaviour.

6.4   Results

First of all, one can study the behaviour of the robot during a simulation from an external point of view. A way to do that is to use our knowledge of the structure of the environment in which the robot lives and build corresponding relevant measures characterizing the behaviour of the robot within a given period of

34

time: 1) the frequency of situations in which it emits a sound within f_1; 2) the frequency of situations in which it emits a sound within f_2; 3) the frequency of situations in which it emits a sound within f_3. Figure 4 shows the evolution of these measures for 5000 time steps. Several phases can be identified:

Stage 1: Initially, the robot produces all kinds of actions with a uniform probability, and in particular produces sounds with frequencies within the whole $[0, 1]$ spectrum.

Stage 1: After the first 250 first steps, the robot concentrates on emitting sounds within f_3, and emits sounds with frequencies within f_1 or f_2 very rarely.

Stage 2: There is then a phase within which the robot concentrates on emitting sounds within f_2, and emits sounds with frequencies within f_1 or f_3 very rarely.

This shows that the robot consistently avoids the situations in which nothing can be learnt, and begins by easy situations and then shifts autonomously to a more complex situation.

We can now study what happens from the robot's point of view. Figure 5 shows a representation of the successive values of $< e_n(t) >$ for all the regions $\mathcal{R}_n$ constructed by the robot at a given time t. As the time is here defined internally as the number of action selection loops, it corresponds to the number of actions that have been chosen by the robot, and to the number of exemplars

that have been provided to it. The graph appears as a tree, which corresponds to the successive splitting of the space into regions. For example, between $t = 0$ and $t = 250$, there is only one curve because during that time there was only one region $\mathcal{R}_1$. This initial curve is the sequence of values of $< e_1(t) >$. Then, because the criterion C_1 was met, this region splits into two regions $\mathcal{R}_2$ and $\mathcal{R}_3$, which also splits the curve into two curves, one corresponding to the successive values of $< e_2(t) >$ and the other corresponding to the successive values of $< e_3(t) >$. Then the curves split again when their associated regions split, etc.

By looking at the trace of the simulation and the definitions of the regions associated to each curve, it is possible to figure out what the regions which are iteratively created look like. It appears that the first split appearing at $t = 250$ corresponds to a split between situations in which the robot emits sounds with a frequency within f_3 ($\mathcal{R}_2$ on the graph), and situations in which the robot emits sounds with a frequency within f_2 or f_3 ($\mathcal{R}_3$ on the graph). To be exact, the system made a split by using the 3rd dimension of $\mathbf{SM(t)}$, i.e. the frequency f, and using the cut value 0.35, which means that the region $\mathcal{R}_2$ includes possibly a small portion of situations with a sound in f_2, since f_2 begins at 0.34^5. Now, we observe that the curve corresponding to $\mathcal{R}_2$ shows a sharp decrease in its error rate, while the curve in $\mathcal{R}_3$ shows an increase in the error rate. This explains why during this period, the robot will emit sounds with frequencies within f_3: indeed, this corresponds to situations which are internally

---

[5]This also shows that the splitting criteria C_1 and C_2 that we presented operate efficiently, since the system finds out by itself that this is the f dimension which is the most relevant for cutting the space at the beginning of the development

evaluated as providing the highest amount of learning progress at this time of its development. Nevertheless, as the robot sometimes does some random actions, the region $\mathcal{R}_3$ accumulates some more exemplars, and we observe that around $t = 320$, it splits into $\mathcal{R}_4$ and $\mathcal{R}_5$. Looking at the trace shows that $\mathcal{R}_4$ corresponds to situations with sounds within f_2 and $\mathcal{R}_5$ with sounds within f_1. We observe that the error rate continues to increase until a plateau is reached for $\mathcal{R}_5$, while it begins to decrease for $\mathcal{R}_4$. During that time, the robot finally predicts perfectly well situations with sounds with a frequency within f_3 and associated with $\mathcal{R}_2$ (it still takes a while because of the noise), and a plateau close to 0 in the error rate is reached. This is why at some point the robot shifts to situations in which it emits sounds with frequencies within f_2, which are situations which are a higher source of learning progress at this point in its development. The robot then tries to vary its motor speeds within this sub-space with sounds with frequencies in f_2, learning to predict how these speeds affect the distance to the toy. The accumulation of new exemplars pushes the robot to split $\mathcal{R}_4$ into more regions, which is a refinement of its categorization of this kind of situations. Now, the system splits the space using the l and r dimensions, and the robot figures out how to explore efficiently the sub-space of situations with sounds with frequencies within f_2, in terms of learning progress.

6.5 Performance in terms of active learning

The efficiency of the exploration of this sub-space of situations with sounds in f_2, where interesting things can be learnt, can be evaluated if we reformulate

IAC within the same problematic as active learning. This will also allow us to evaluate the efficiency of the IAC algorithm from the point of view of active learning. Indeed, as we explained in the introduction, in the field of machine learning and data mining, the search for methods which allow to reduce the number of examples needed to achieve a given level of performance in generalization for a machine which learns an input-output mapping, is of growing interest (here the input is $\mathbf{SM(t)}$ and the output is $SM(t+1)$). While IAC was designed as a system for driving the development of a robot, it can also be considered as a pure active learning algorithm, and in this respect it is interesting to evaluate how it compares with standard existing algorithms. Thus, we will use two reference algorithms to evaluate the performance of IAC. The first one follows the most common idea in the field of active learning ([25], [15], [24]): the choice of the next action (also called query or experiment depending on the authors) is done such that it corresponds to an input-output pair for which the machine evaluates that its prediction for this pair will be maximally false as compared to its prediction for possible other pairs. It is easy to adapt this idea using the same algorithmic architecture than the one used for IAC: when the robot has to decide for an action in a given context, it makes the list of possible actions within that context, then for each of them evaluates the expected error in prediction using the quantity $< e_{mean}(t) >$ defined earlier, and finally chooses the action for which this quantity is maximal. Everything else is equal. We will call this algorithm "MAX". The second reference algorithm that we use is the "RANDOM" algorithm, which simply consists in random action selection

(and so is not an active learning algorithm, but serves as a baseline).

IAC, MAX and RANDOM will be compared in terms of their performance in generalization in predicting the consequence of actions characterized by a frequency within the f_2 zone. This means that we will evaluate each of them in the part which we know is interesting. Yet, the whole space with all ranges of frequencies is made available to the robot, which does not know as earlier that there is a particular zone where it can actually learn non trivial things.

For a given simulation using a given algorithm among IAC, MAX and RANDOM, we evaluate every 100 actions the performance in generalization of the current learning machine. To do that, we initially made a simulation with random action selection and collected a database of input-output by storing the experienced $(\mathbf{SM(t)}, \mathbf{S(t+1)})$ couples for which the action included an emission of a sound with a frequency within f_2. This provides an independent test set which we used to test the capacity of prediction that the robot acquired at a given time in its development. For this test which is done every 100 actions, we freeze the learning machine and make it predict the output corresponding to all the inputs in the test database. The freezing ensures that the machine does not learn while it is tested. The prediction accuracy is measured using the mean squared error over the database. After evaluating the performance, we unfreezed the system until the next evaluation.

Figures 6 shows typical resulting curves of the three algorithms. We see that initially, the algorithm which learns fastest is the RANDOM algorithm. This is normal since MAX spends times in uninteresting situations, and IAC at the

39

beginning spends time in the easy situation, so RANDOM is the algorithm which provides initially the highest amount of examples related to the production of the sounds with frequencies within f_2 (33 percent of examples are of this type in this case). Then, after 3000 actions, the curve corresponding to the IAC algorithm suddenly drops down: this corresponds to the shift of attention of the robot towards situations with sounds with frequencies within f_2. Now, this robot spends 85 percent of its time in situations with sound with frequency within f_2 (and not 100 percent due to the 0.15 probability to do a random action). Quickly, the curve gets significantly below the RANDOM algorithm, and reaches a low plateau around 5000 actions (where the mean prediction error stays around 0.09). The RANDOM curve reaches a low plateau much later (this is not represented on this curve) after about 11000 actions. The value of the plateau, interestingly, is higher than with the IAC algorithm: it is 0.096. We repeated 100 times the experiments in order to see whether this had some statistical significance. In each simulation, we measured the time where a plateau was reached (defined as 500 successive points where the mean squared error has a variance smaller than 0.0001), and what the mean squared error was at that time. It turned out that the plateau was reached at $t = 4583$ in average for IAC, with a standard deviation of 452, and at $t = 11980$ in average with a standard deviation of 561 for RANDOM. The mean squared error was $e = 0.89$ in average with a standard deviation of 0.009 for IAC, and was $e = 0.96$ with a standard deviation of 0.004 for RANDOM. As a consequence, we can say consistently that IAC allows to learn the interesting part of the mapping

40

about 2.6 times faster and with a higher performance in generalization than the RANDOM algorithm. This increase of the performances in generalization is similar to what has already been described in other active learning algorithm ([32]).

6.6 Summary

With this experiment we have shown a first embodiment of the IAC system within a simulated robot. This has allowed us to show how IAC could manage the development of the robot in an inhomogeneous sensorimotor environment with parts which were not learnable by the robot. We have shown how the robot consistently avoided this zone of unlearnability and on the other hand explored autonomously sensorimotor situations of increasing complexity. This simple set-up also allowed us to detail the evolution of the internal structures built by the IAC system. We could explain for example the progressive formation of regions with varying potentials for learning progress. Finally, this set-up not only allowed us to show the interest of IAC as an intrinsic motivation system which could self-organize the behaviour of a robot in a developmental manner, but also it showed that IAC is an efficient and robust active learning system. Indeed, we proved that it was faster than both the RANDOM algorithm and traditional active learning methods which are not suited to mappings with strong inhomogeneities and even unlearnable parts.

Yet, the simplicity of this set-up did not allow to show how a developmental sequence with more than one transition could self-organize autonomously (here,

there was only a transition between a stage in which the robot focused on actions with sounds in f_1, and then a stage in which the robot focused on actions with sounds in f_2). We are now going to present a more complex experiment in which we will show that multiple sequential levels of self-organization of the behaviour of the robot can happen.

7 The Playground Experiment: the discovery of sensorimotor affordances

This new experimental set-up is called "The Playground Experiment". This involves a physical robot as well as a more complex sensorimotor system and environment. We use a Sony AIBO robot which is put on a baby play mat with various toys that can be bitten, bashed or simply visually detected (see figure 7). The environment is very similar to the ones in which two or three month old children learn their first sensorimotor skills, although the sensorimotor apparatus of the robot is here much more limited. We have developed a web site which presents pictures and videos of this set-up: http://playground.csl.sony.fr/.

7.1 Motor control

The robot is equipped initially only with simple motor primitives. In particular it is not able to walk around. There are three basic motor primitives: turning the head, bashing and crouch biting. Each of them is controlled by a number of real number parameters, which are the action parameters that the robot controls.

The "turning head" primitive is controlled with the pan and tilt parameters of the robot's head. The "bashing" primitive is controlled with the strength and the angle of the leg movement (a lower-level automatic mechanism takes care of setting the individual motors controlling the leg). The "crouch biting" primitive is controlled by the depth of crouching (and the robot crouches in the direction in which it is looking at, which is determined by the pan and tilt parameters). To summarize, choosing an action consists in setting the parameters of the 5-dimensional continuous vector $\mathbf{M(t)}$:

$$\mathbf{M(t)} = (p,\ t,\ b_s,\ b_a,\ d)$$

where p is the pan of the head, t the tilt of the head, b_s the strength of the bashing primitive, b_a the angle of the bashing primitive, and d the depth of the crouching of the robot for the biting motor primitive. All values are real numbers between 0 and 1, plus the value -1 which is a convention used for not using a motor primitive: for example, $\mathbf{M(t)} = (0.3, 0.95, -1, -1, 0.29)$ corresponds to the combination of turning the head with parameters $p = 0.3$ and $t = 0.95$ with the biting primitive with the parameter $d = 0.29$ but with no bashing movement.

7.2   Perception

The robot is equipped with three high-level sensors based on lower-level sensors. The sensory vector $\mathbf{S(t)}$ is thus 3-dimensional:

$$\mathbf{S(t)} = (O_v,\ B_i,\ O_s)$$

where:

- O_v is the binary value of an object visual detection sensor: It takes the value 1 when the robot sees an object, and 0 in the other case. In the playground, we use simple visual tags that we stick on the toys and are easy to detect from the image processing point of view. These tags are black and white patterns similar to the Cybercode system developed by Rekimoto ([43]).

- B_i is the binary value of a biting sensor: It takes the value 1 when the robot has something in its mouth and 0 otherwise. We use the cheek sensor of the AIBO;

- O_s is the binary value of an oscillation sensor: It takes the value 1 when the robot detects that there is something oscillating in front of it, and 0 otherwise. We use the infra-red distance sensor of the AIBO to implement this high-level sensor. This sensor can detect for example when there is an object that has been bashed in the direction of the robot's gaze, but can also detect events due to human walking around the playground (we do not control the environment).

It is crucial to note that initially the robot knows nothing about sensorimotor affordances. For example, it does not know that the values of the object visual detection sensor are correlated with the values of its pan and tilt. It does not

44

know that the values of the biting or object oscillation sensors can become 1 only when biting or bashing actions are performed towards an object. It does not know that some objects are more prone to provoke changes in the values of the B_i and O_s sensors when only certain kinds of actions are performed in their direction. It does not know for example that to get a change in the value of the oscillation sensor, bashing in the correct direction is not enough, because it also needs to look in the right direction (since its oscillation sensors are on the front of its head). These remarks allow to understand easily that a random strategy will not be efficient in this environment. If the robot would do random action selection, in a vast majority of cases nothing would happen (especially for the B_i and O_s sensors).

7.3   The action perception loop

To summarize, the mapping that the robot has to learn is:

$$f : \ \mathbf{SM(t)} = (p, \ t, \ b_s, \ b_a, \ d, \ O_v, \ B_i, \ O_s)$$

$$\longmapsto \mathbf{S(t+1)} = (\widetilde{O_v}, \ \widetilde{B_i}, \ \widetilde{O_s})$$

The robot is equipped with the Intelligent Adaptive Curiosity system, and thus chooses its actions according to the potential learning progress that it can provide to one of its expert. In this experiment, the action perception loop is rather long: when the robot chooses and executes an action, it waits that all its motor primitives have finished their execution, which lasts approximately one second,

before choosing the next action. This is how the internal clock for the IAC system is implemented. On the one hand, this allows the robot to make all the measures necessary for determining adequate values of (O_v, B_i, O_s). On the other hand and most importantly, this allows the environment to come back to its "resting state". This means that environment has no memory: after an action has been executed by the robot, all the objects are back in the same state. For example, if the object that can be bashed has actually been bashed, then it has stopped oscillating before the robots tries a new action. This is a deliberate choice to have an environment with no memory: while keeping all the advantages, the constraints and the complexity of a physical embodiment, this makes that mapping from actions to perception learnable in a reasonable time. This is crucial if one wants to do several experiments (already in this case, each experiment lasts for nearly one day). Furthermore, introducing an environment with memory frames the problem of the maximization of internal reward within delayed reward reinforcement problems, for which there exists powerful but complicated techniques whose biases would certainly make the results more complex and render them more difficult to interpret.

7.4 Results

During an experiment we continuously measure a number of features which help us characterize the dynamics of the robot's development. First, we measure the frequency of the different kinds of actions that the robot performs in a given time window. More precisely:

- the percentage of actions which do not involve the biting and the bashing motor primitive in the last 100 actions (i.e. the robot's action boils down to "just looking" in a given direction).

- the percentage of actions which involve the biting motor primitive in the last 100 actions.

- the percentage of actions which involve the bashing motor primitive;

Then, we track the gaze of the robot and at each action measure if it is looking towards 1) the bitable object, or 2) the bashable object, or 3) no object. This is possible since from an external point of view we know where the object are and so it is easy to derive the information from the head position.

Third, we measure the evolution of the frequency of successful biting actions and the evolution of successful bashing actions. A successful biting action is defined as an action which provokes a "1" value on the B_i sensor (an object has actually be bitten). A successful bashing action is defined as an action which provokes an oscillation in the O_s sensor.

Figure 8 shows an example of result, showing the evolution of the three kinds of measures on three different levels. A striking feature of these curves is the formation of sequences of peaks. Each of these peaks means basically that at the moment it occurs the robot is focusing its activity and its attention on a small subset of the sensorimotor space. So it is qualitatively different from random action performance in which the curves would be stationary and rather flat. By looking in details at these peaks and at their co-occurence (or not) within the

different kinds of measures, we can make a description of the evolution of the robot's behaviour. On figure 8, we have marked a number of such peaks with letters from A to G. We can see that before the first peak, there is an initial phase during which all actions are produced equally often, that most often no object is seen, and that a successful bite or bash only happens extremely rarely. This corresponds to a phase of random action selection. Indeed, initially the robot categorizes the sensorimotor space using only one big region (and so there is only one category), and so all actions in any contexts are equally interesting. Then we observe a peak (A) in the "just looking" curve: this means that for a while, the robot stops biting and bashing, and focuses on just moving its head around. This means that at this point the robot has split the space into several regions, and that a region corresponding to the sensorimotor loop of "just looking around" is associated to the highest learning progress from the robot's point of view. Then, the next peak (B) corresponds to a focus on the biting action primitive (with various continuous parameters), but it does not co-occur with the looking towards the bitable object. This means that the robot is trying to bite basically in all directions around him : it did not discover yet the affordances of the biting actions with particular objects. The next peak (C) corresponds to a focus on the bashing action primitive (with various continuous parameters) but again the robot does not look towards a particular direction. As the only way to discover that a bashing action can make an object move is by looking in the direction of this object (because the IR sensor is on the cheek), this means that the robot does not use at this point the bashing primitive with

the right affordances. The next peak (D) corresponds to a period within which the robot stops again biting and bashing and concentrates on moving the head, but this times we observe that the robot focuses these "looking" movement in a narrow part of the visual field : it is basically looking around one of the objects, learning how it disappears/reappears in its field of view. Then, there is a peak (E) corresponding to a focus on the biting action, which is this time coupled with a peak in the curve monitoring the looking direction towards the bitable object, and a peak in the curve monitoring the success in biting. It means that during this period the robot uses the action primitive with the right affordances, and manages to bite the bitable object quite often. This peak is then repeated a little bit later (F). Then finally a co-occurrence of peaks (G) appears that corresponds to a period during which the robot concentrates on using the bashing primitve with the right affordances, managing to actually bash the bashable object quite often.

This example shows that several interesting phenomena have appeared in this run of the experiment. First of all, the presence and co-occurrence of peaks of various kinds shows a self-organization of the behavior of the robot, which focuses on particular sensorimotor loops at different periods in time. Second, when we observe these peaks, we see that they are not random peaks, but show a progressive increase in the complexity of the behaviour to which they correspond. Indeed, one has to remind that the intrinsic dimensionality of the "just looking" behaviour (pan and tilt) is lower than the "biting" behaviour (which adds the depth of the crouching movement), which is itself lower than

the "bashing" behaviour (which adds the angle and the strength dimensions). The order of appearance of the periods within which the robot focuses on one of these activities is precisely the same. If we look in more details, we also see that the biting behaviour appears first in a non-affordant version (the robot tries to bite things which cannot be bitten), and then only later in the affordant version (where it tries to bite the biteable object). The same observation holds for the bashing behaviour: first it appears without the right affordances, and then it appears with the right affordances. The formation of focused activities whose properties evolve and are refined with time can be used to describe the developmental trajectories that are generated in terms of stages: indeed, one can define that a new stage begins when a co-occurence of peaks that never occured before happens (and so which denotes a novel kind of focused activity).

We ran several times the experiment with the real robots, and whereas each particular experiment produced curves which were different in the details, it seemed that some regularities in the patterns of peak formation, and so in terms of stage sequences, were present. We then proceeded to more experiments in order to assess precisely the statistical properties of these self-organized developmental trajectories. Because each experiment with the real robot lasts several hour, an in order to be able to run many experiments (200), we developed a model of the experimental set-up. Thanks to the fact that the physical environment was memoryless after each action of the robot, it was possible to make an accurate model of it using the following procedure: we let the robot perform several thousands actions and we recorded each time $SM(t)$ and $S(t+1)$. Then,

from this database of examples we trained a prediction machine based on locally weighted regression [44]. This machine was then used as a model of the physical environment and the IAC algorithm of the robot was directly plugged into it.

Using this simulated world set-up, we ran 200 experiments, each time monitoring the evolution using the same measures as above. We then constructed higher-level measures about each of the runs, and based on the structure of the peak sequence. Peaks where here defined using a threshold on the height and width of the bumps in the curves. These measures correspond to the answer to these following questions:

- (Measure 1) **number of peaks?**: How many peaks are there in the action curves (top curves) ?

- (Measure 2) **complete scenario?**: Is the following developmental scenario matched: first there is a "just looking" peak, then there is a peak corresponding to "biting" with the wrong affordances which appears before a peak corresponding to "biting" with the right affordances, and there is a peak corresponding to "bashing" with the wrong affordances which appears before a peak corresponding to "bashing" with the right affordance (and the relative order between "biting"-related peaks and "bashing"-related peaks is ignored). Biting with the right affordance is here defined as the co-occurence between a peak in the "biting" curve and a peak in the "seeing the biteable object" curve, and biting with the wrong affordances is defined as all other situations. The corresponding definition applies to "bashing".

- (Measure 3) **nearly complete scenario?**: Is the following less constrained developmental scenario matched: there is a peak corresponding to "biting" with the wrong affordances which appears before a peak corresponding to "biting" with the right affordances, and there is a peak corresponding to "bashing" with the wrong affordances which appears before a peak corresponding to "bashing" with the right affordances (and the relative order between "biting"-related peaks and "bashing"-related peaks is ignored).

- (Measure 4) **non-affordant bite before affordant bite?**: Is there is a peak corresponding to "biting" with the wrong affordances which appears before a peak corresponding to "biting" with the right affordances?

- (Measure 5) **non-affordant bash before affordant bash?**: there is a peak corresponding to "bashing" with the wrong affordances which appears before a peak corresponding to "bashing" with the right affordances?

- (Measure 6) **period of systematic successful bite?** Does the robot succeeds systematically in biting often at some point (= is there a peak in the "successful bite" curve)?

- (Measure 7) **period of systematic successful bash?** Does the robot succeeds systematically in bashing often at some point (= is there a peak in the "successful bash" curve?

- (Measure 8) **bite before bash ?** Is there a focus on biting which appears before a focus on bashing (independantly of affordance) ?

- (Measure 9) **successful bite before successful bash?** Is there a focus on successfully biting which appear before a focus on successfully bashing ?

The numerical results of these measures are summarized in table 1. This table shows that indeed some structural and statistical regularities arise in the self-organized developmental trajectories. First of all, one has to note that the complex and structured trajectory described by Measure 2 appears in 34 percent of the cases, which is high given the number of possible co-occurences of peaks which define a combinatorics of various trajectories. Furthermore, if we remove the test on "just looking", we see that in the majority of experiments, there is a systematic sequencing from non-affordant to affordant actions for both biting and bashing. This shows an organized and progressive increase in the complexity of the behaviour. Another measure confirms this increase of complexity from another point of view: if we compare the relative order of appearance of periods of focused bite or bash, then we find that "focused bite" appears in the large majority of the cases before the "focused bash", which corresponds to their relative intrinsic dimension (3 for biting and 4 for bashing). Finally, one can note that the robot reaches in 100 percent of the experiments a period during which it repeatedly manages to bite the biteable object, and in 78 percent of the experiments it reaches a period during which it repeatedly manages to bash the bashable object. This last point is interesting since the robot was not pre-programmed to achieve this particular task.

These experiments show how the intrinsic motivation system which is implemented (IAC) drives the robot into a self-organized developmental trajectory in which periods of focused sensorimotor activities of progressively increasing complexity arise. We have seen that a number of structural regularities arose in the system, such as the tendancy of non-affordant behaviour to be explored before affordant behaviour, or the tendancy to explore a certain kind of behaviour (bite) before another kind (bash). Yet, one has also to stress that these regularities are only statistical: two developmental trajectories are never exactly the same, and more importantly it happens that some particular trajectories observed in some experiments differ qualitatively from the mean. Figure 9 illustrate this point. The figures on the top-left and top-right corners presents runs which are very typical and corresponds to the "complete scenario" described by Measure 1. On the contrary, the runs presented on the bottom-left and bottom-right corners corresponds to atypical results. The experiment of which curves are presented in the bottom-left corner shows a case where the focused exploration of bashing was performed before the focused exploration of biting. Nevertheless, in this case the regularity "non-affordant before affordant" is preserved. On the bottom-right corner, we observe a run in which the affordant bashing activity appears very early and before any other focused activity. This balance between statistical regularities and diversity has parallels in infant sensorimotor development [45]: there are some strong structural regularities but from individual to individual there can be some substantial differences (for e.g. some infants learn how to crawl before they can sit and other do the reverse).

8 Discussion

8.1 Developing complex behavioural schemas

We have discussed how to design a source of internal rewards suited for active and autonomous development. Such an intrinsic motivation system permits to realize an efficient active exploration of a given sensorimotor space. In the experiments described, we deliberately considered simple spaces. Enhancing the complexity of perception and motor spaces seems crucial in order to expect the emergence of more complex forms of behaviour. However, designing suitable spaces that can lead to complex behavioural patterns raises several difficult issues.

A first issue is whether perception and motor spaces should be considered as two independent spaces. The intrinsic links that bind perception with action have been stressed by many authors. In some circumstances, relevant information about a given environment arises from sensorimotor trajectories rather than from simple analysis of perceptual data. Several experiments have shown that agents can simplify problems of categorizing situations by actively modifying their own position or orientation with respect to the environment or by modifying the environment itself. In the same manner, certain environmental regularities can be detected only by producing particular stereotyped behaviour (e.g. [46, 47]). The fact that perception is fundamentally active, naturally leads to consider abstractions like *behavioural schemas* as relevant unit for understanding development.

Schemas are famously known as central elements of Piaget's developmental psychology but the term has also been used in neurology, cognitive psychology and motor control ([48] p.36–40) and related notions appeared in artificial intelligence under names like *frames* or *scripts* [49, 50]. In Piaget's theory, children's development can be interpreted as the incremental organization of a set of schemas. Schemas are skills that serve both for perceiving the environment and acting upon it. Piaget calls *assimilation* the ability to make sense of a situation in terms of a current set of schemas and *accommodation* the way in which schemas are updated as the expectations based on assimilation are not met. The child starts with basic sensorimotor schemas such as suckling, grasping and some primary forms of eye-hand coordination. Through accommodation and assimilation, new schemas are created, and sets of existing schemas get coordinated. The child makes progressively more complex abstract inferences about the environment, leading eventually to language and logic, forms of abstract thought that are no longer directly grounded in particular sensorimotor situations. The whole developmental trajectory can be interpreted as an extension from a simple sensorimotor space to an elaborated mental space. The space changes but the fundamental dynamics of accommodation and assimilation that actively drive the child's behaviour remain the same.

It is important to stress that schemas are primarily *functional* units. In that sense, they are *a priori* distinct from structural units that can be identified in the organization of the organism or the machine that produces the observed behaviour. However, many artificial intelligence models make use of internal

explicit schema structures. In such systems, there is a one-to-one mapping between these internal structures and the functional operation that the agent can perform. For instance, Drescher describes a system inspired by Piaget's theories in which a developing agent explicitly creates, modifies and merges schema structures in order to interact with a simple simulated environment [51]. Using explicit schema structures has several advantages: such structures can be manipulated via symbolic operations, creation of new skills can be easily monitored by following the creation of new schemas, etc.

Other systems do not rely on such explicit representations. These are typically subsymbolic systems, using continuous representations of their environment. Nevertheless, such systems may display some organized forms of behaviour where clear functional units can be identified. Their developmental trajectories can also be interpreted as a progressive organization of schemas. For instance, the developmental trajectories produced by the typical experiments of section VII can be interpreted as assimilation and accommodation phases. In these typical runs, the robot "discovers" the biting and bashing schema by producing repeated sequences of these kinds of behaviour, but initially these actions are not systematically oriented towards the biteable or the bashable object. This stage corresponds to "assimilation". It is only later that "accommodation" occurs as biting and bashing starts to be associated with their respective appropriate context of use. Our experiments show that functional organization can emerge even in the absence of explicit internal schema structures. However, the current limitations of such a system may appear when considering more complex

forms of behavioural organization such as formation of hierarchical structures and the emergence of goals.

8.1.1 Hierarchical organization

Complex behavior patterns are hierarchically organized. For instance, a complex motor program is often described as an abstract event sequence at a high level and a detailed motor program in a lower level. Therefore, possibility for forming level structures is a key issue. Different authors have already tried to tackle how combinations of primitives could be autonomously organized in higher level structures. *Option theory* offers an interesting mathematical framework to address hierarchical organization of systems using explicit schema structures [52]. Options are like subroutines associated with closed-loop control structures. They can invoke other options as components. Barto, Singh and Chentanez have recently illustrated in a simple environment how options could be used to develop a hierarchical collection of skills [21]. Hierarchical organization of explicit schemas is also illustrated by the work of Drescher among others [51]. But, can hierarchically-organized behavior appear in the absence of explicit schemas? Different attempts have been made in this direction. A multiple model-based reinforcement learning capable of decomposing a task based on predictability levels was proposed by Doya, Samejima, Katagiri and Kawato [53]. Tani and Nolfi presented a system capable of combining local experts using gated modules [54]. However, in all these studies explicit level structures

are predetermined by the network architecture. The question whether hierarchical structures can simply self-organize without being explicitly programmed remains open.

8.1.2 Goal-directedness

Complex behavior patterns are also associated with intentionally directed processes. This means that they are performed by an agent trying to achieve a particular desirable situation that constitutes its aim or *goal* (e.g. reducing hunger, following someone, learning something). The agent's behavior reflects his or her *intention*, that is the plan of action that the agent chooses for realizing this particular goal. This plan includes both the means and the pursued goal [55]. Once again, systems using explicit schema structure embed these notions of goals and means as explicit symbolic representations. Such explicit goals can be created, updated, deleted and more importantly easily monitored. This has led to numerous systems in classical artificial intelligence, and research in this area has influenced importantly the way we consider decision making or planning. More recently, research on agent architectures [56] has put a major emphasis on the same issues. However, these models do not give much insight on the developmental and cognitive mechanisms that lead to the notion of intentionally-directed behaviour. Can goals and means simply emerge out of subsymbolic dynamics? This is one of the most challenging issue developmental approaches to cognition have to face [57]. To some extent, certain reinforcement

learning models have demonstrated that the organization of behavior into goals and subgoals can be interpreted as emergent features resulting of simpler drives [37]. But no subsymbolic systems currently matches the performances and the flexibility of systems using explicit goal-directed schemas.

8.1.3 Generalization, transfer, analogy

Generalization, transfer or analogies between schemas are also thought to be central for the emergence of complex behavior patterns (see [58] for a general discussion of the issue of transfer in cognition). Skills do not develop independently from one another. The ones that have structural relationship bootstrap each other. In particular, processes of analogy and metaphors are crucial for transferring know-how developed in sensorimotor contexts to more abstract spaces [59]. There is an important literature on how to compare explicit schema structure (e.g. [60]), but many authors have argued that generalization and transfer of skills could also be (maybe even more) efficient in the absence of symbolic representation [61]. This debate bears some resemblance with the opposition between localists or distributed kinds of representation. Systems with explicit schema structures, but also many subsymbolic systems using memories organized into local structures (e.g. sets of experts) are called localists. In this scheme, learning a new behavior schema corresponds to the addition of a template to an existing set of modules. The independence of the modules facilitates incremental learning as each addition do not cause interferences

with the existing memory contents. However, extension to unknown patterns must be realized with ad-hoc processes that specify the way similarity should be computed. In the same manner, generalization across a large set of local representations is intrinsically difficult. On the contrary, in systems with distributed representations, behavior schemas are not assigned to particular modules but are memorized in a distributed manner (e.g. as synaptic weights of global neural network). This means that each schema can only exist in relation to others. Self-organized generalization processes are facilitated in such context [62].

Developmental trajectories of intrinsically motivated agents are constrained by many factors. We have briefly discussed some of the important issues for designing systems capable of developing reusable, goal-directed, hierarchically-organized behavioural schemas. Investigating the dynamics resulting of the intrinsic motivation systems embedded in such kinds of more complex spaces will be the topic of future research.

8.2 Relation to developmental psychology

Our research takes clear inspiration from developmental psychology both conceptually (the notion of intrinsic motivation originally comes from psychology) and methodologically (analysis of the development in terms of qualitative sequences of different kinds of behavioural patterns). Could our model be interesting in return for interpreting processes underlying infant's development? More precisely:

- Can we interpret particular developmental processes as being the result of a *progress drive*, an intrinsic motivation system driving the infant into situations expected to result in maximal learning progress?

- Can operant models of intrinsic motivation provide useful abstraction that address the complexity of infant's development?

Some initial attempts have been taken to start answering these questions. Taking ground on preliminary experimental results, we discussed in [63] a scenario presenting the putative role of the progress drive for the development of early imitation. We argue in particular that progress-driven learning could help understanding why children focus on specific imitative activities at a certain age and how they progressively organize preferential interactions with particular entities present in their environment.

8.2.1 Progress niches

To facilitate interpretation, we introduced the notion of *progress niches* to characterize the behaviour of our model. The progress drive pushes the agent to discover and focus on situations which lead to maximal learning progress. These situations, neither too predictable nor too difficult to predict, are "progress niches". Progress niches are not intrinsic properties of the environment. They result from a relation between a particular environment, a particular embodiment (sensors, actuators, feature detectors and techniques used by the prediction algorithms) and a particular time in the developmental history of the agent. Once discovered, progress niches progressively disappear as they become

more predictable. The notion of progress niches is related to Vygotsky's *zone of proximal development*, where the adult deliberately challenges the child's level of understanding. Adults push children to engage in activities beyond their current mastery level, but not too far beyond so that they remain comprehensible [64]. We could interpret the zone of proximal development as a set of potential progress niches organized by the adult in order to help the child learn. But it should be clear that independently of the adults' efforts, what is and what is not a progress niche is ultimately defined from the child's point view. Progress niches share also similarities with Csikszentmihalyi's *flow experiences* [8]. Csikszentmihalyi argues that some activities are *autotelic* when challenges are appropriately balanced with the skills required to cope with them (see also [65]). We prefer to use the term progress niche by analogy with ecological niches as we refer to a transient state in the evolution of a complex "ecological" system involving the embodied agent and its environment.

8.2.2 Self-other distinction

Using this terminology, the computational model presented in this paper shows how an agent can (1) separate its sensorimotor space into zones of different predictability levels and (2) choose to focus on the one which leads to maximal learning progress, called a "progress niche". With this kind of operant models, it could be speculated that meaningful sensorimotor distinctions (self, others and objects in the environment) may be the result of discriminations constructed

during a progress-driven process. We can more specifically offer an interpretation of several fundamental stages characterizing infant's development during their first year.

- **Stage 1: Like-me stance (0-1m)**. Simple forms of imitative behaviour have been argued to be present just after birth. They could constitute a process of early identification. Some totally or partially nativist explanations could account for this early "like-me stance" [66, 67]. This would suggest the possibility of an early distinction between persons and things. If an intermodal mapping facilitating the match between what is seen and what is felt exists, the hypothesis of a progress drive would suggest that infants will indeed create a discrimination between such easily predictable couplings (interaction with peers) and unpredictable situations (all the other cases) and that they will focus on the first zone of their sensorimotor space that constitutes a "progress niche". Neonates imitation (when it occurs) would be the result of the exploitation of the most predictable coupling present just after birth.

- **Stage 2: Circular reactions (1-2m)**. During the first two months of their life, infants perform repeated body motion. They kick their legs repeatedly, they wave their arms. This process is sometimes referred as "body babbling". However, nothing indicates that this exploratory behaviour is randomly organised. Rochat argues that children are in fact performing self-imitation, trying to imitate themselves [68]. This would mean that children are structuring their own behaviour in order to make

64

it more predictable and form this way "circular reactions" [69, 41]. Such self-imitative behaviours can be well explained by the progress drive hypothesis. Sensorimotor trajectories directed towards the child's own body can be easily discriminated from trajectories directed towards other people by comparing their relative predictability difficulty. By many respects, making progress in understanding primary circular reactions is easier than in the cases involving other agents: Self-centered types of behaviour are "progress niches". In such a scenario the "self" emerges as a meaningful discrimination for achieving better predictability. Once this distinction is made, progress for predicting the effects of self-centered actions can be rapidly made.

- **Stage 3: Self-other interactions (2-4m)**. After two months, infants become more attentive to the external world and particularly to people. Parental scaffolding plays a critical role for making the interaction with the child more predictable [70]. Parents adapt their own responses so that interactions with the child follow the normal social rules that characterize communicative exchanges (e.g. turn taking). Moreover, if an adult imitates an infant's own actions, it can trigger continued activity in the infant. This early imitative behaviour is referred as "pseudo-imitation" by Piaget [71]. Pseudo-imitation and focus on scaffolded adult behaviour could be seen as predictable effects of the progress drive. As the self-centered trajectories start to be well mastered (and do not constitute "progress niches" anymore), the child's focus shifts to another branch of the discrimination

tree, the "self-other" zone.

- **Stage 4: Interactions with objects (5-7m)**. After five months, attention shifts again from people to objects. Children gain increased control over the manipulation of some objects on which they discover "affordances" [72]. Parents recognize this shift and initiate interactions about those affordant objects. However, children do not alternate easily their attention between the object and their caregiver. A progress-driven process can account for this discrimination between affordant objects and unmastered aspects of the environment. Although this stage is typically not seen as imitative, it could be argued that the exploratory process involved in the discovery of the object affordances shares several common features with the one involved for self-centered activities: the child structures its world looking for "progress niches".

We have to stress that the system discussed in this paper is not meant to re-enact precisely infant's developmental sequence, and is not a model of human development. For instance, the playground experiment focuses directly on the discovery of object's affordances. Yet, in addition to the developmental robotics engineering techniques that it explores, we think that this system, as well as other existing intrinsic artificial intrinsic motivation systems, can also be used as a "tool for thoughts" in developmental psychology. In that sense, it may help formulating new concepts useful for the interpretation of the developmental dynamics underlying children's development. For example, the existence of a *progress* drive could explain why certain types of imitative behaviour are

produced by children at a certain age and stop to be produced later on. It could also explain how discrimination between actions oriented towards the self, towards others and towards the environment may occur. However, we do not even imagine that a drive for maximizing learning progress could be the only motivational principle driving children's development. The complete picture is likely to include a complex set of drives. Developmental dynamics are certainly the result of the interplay between intrinsic and extrinsic forms of motivations, particular learning biases, as well as embodiment and environmental constraints. We believe that computational and robotic approaches can help specifying the contribution of these different components in the overall observed patterns and shed new light on the particular role played by intrinsic motivation in these complex processes.

9 Conclusion

Intrinsic motivation systems are likely to play a pivotal role for the future of developmental robotics. In this paper, we have presented the background in developmental psychology, neuroscience, and machine learning. We showed that current efforts in the developmental robotics community are approaching the construction of intrinsic motivation system through the operationalization and implementation of concepts such as "novelty", "surprise" or more generally "curiosity". We have reviewed some representative works in this direction, trying to classify them into different groups according to the way they operationalized

curiosity. Then we presented an intrinsic motivation system called Intelligent Adaptive Curiosity, which was conceived to drive the development of a robot in continuous noisy inhomogeneous environmental and sensorimotor spaces, permitting an autonomous self-organization of behavior into a developmental trajectory with sequences of increasingly complex behavioural patterns. This was made possible thanks to the way the system evaluates its own learning progress, through the combination of a regional evaluation of the similarity of situations with a smoothing of the error rate curves associated to each region.

This system was tested in two robotic set-ups. In a first simple simulated robotic set-up, we showed in detail how the system works, and provokes both behavioural and cognitive development, by looking in details into the traces of the simulation. This first set-up also showed how IAC can allow a robot to avoid situations which are not learnable by the system, and engage in situations of progressively increasing complexity in terms of difficulty of learning, which leads to a self-organization of the behaviour. This first set-up also allowed to show that our intrinsic motivation system could be used efficiently as an active learning algorithm robust in inhomogeneous spaces. Some currently ongoing work suggests that these results still hold in high-dimensional continuous spaces. If this is confirmed, this would allow to attack real-world learning problems whose properties of inhomogeneity kept them out of reach of standard active learning methods so far [33]. In a second real and more complex robotic set-up, we showed how IAC can drive the development of a robot through more than one developmental transition, and thus allows the robot to generate autonomously a

developmental sequence. Doing these experiments was also the opportunity to discuss methodological issues related to the evaluation of a developmental robot. Indeed, classical machine learning methods of evaluation, based on the measure of the performance of a system on a given human-defined task, are not suited for developmental robots since one of their key features is to be task-independent, as advocated by Weng ([34]). We explained that a developmental evaluation should be based on the monitoring of the evolution of the complexity of the system from different points of view, since indeed complexity is a concept which is observer-dependent. For example, it is a necessity to couple a measure of the evolution of the complexity from the robot's point of view, and the monitoring of its behavior on a long time scale using methods inspired from human sciences and developmental psychology.

We have also discussed the limits of the system as we presented it in this paper. Indeed, there are two kinds of limitations which will be the subject of future work. On the one hand, we deliberately made the simplification that what the system should optimise is the immediate reward ($r(t + 1)$). This allowed us not to use complex re-inforcement techniques and limit the biases coming from the action selection procedure in order to better understand the properties of our learning progress measure. Nevertheless, this will be a necessity in the future to use such complex re-inforcement learning techniques, since in the real world progress niches are not always readily accessible, and thus comes the problems of delayed rewards. This extension of our system should certainly be inspired by the work of Barto, Singh and Chentannez ([21]) who have presented

69

a study which is very complementary to ours, in which they experimented the use of a complex re-inforcement technique given a simple novelty-based intrinsic motivation system.

A second kind of limitation which characterizes the current system is the fact that the sensorimotor space is rather simple, in particular from the point of view of representation. It is an open issue to study how forms of representations more complex than scalar vectors, such as schemas for example, could be integrated within the Intelligent Adaptive Curiosity system. One of the potential problems to be solved is if several levels of representations are used: how can one build measures of learning progress or knowledge gain which are homogeneous and allow the comparison of activities or sensorimotor contexts which involve different representations?

Finally, we have seen that even if the primary goal of the system we presented is to allow the construction of a truly developmental robot, taking inspiration from human development, the system could in return possibly be useful for developmental psychologists as a tool for thoughts. Indeed, we explained how it can help to formulate new concepts for the interpretation of the developmental dynamics involved in human infant's development.

10 Acknowledgements

References

[1] 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, pp. 599–600, 2001.

[2] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: A survey," *Connection Science*, vol. 15, no. 4, pp. 151–190, 2003.

[3] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda, "Purposive behavior acquisition on a real robot by vision-based reinforcement learning," *Machine Learning*, vol. 23, pp. 279–303, 1996.

[4] J. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, pp. 71–99, 1993.

[5] R. White, "Motivation reconsidered: The concept of competence," *Psychological review*, vol. 66, pp. 297–333, 1959.

[6] E. Deci and R. Ryan, *Intrinsic Motivation and Self-Determination in Human Behavior.* Plenum Press, 1985.

[7] D. Berlyne, *Conflict, Arousal and Curiosity.* McGraw-Hill, 1960.

[8] M. Csikszenthmihalyi, *Flow-the psychology of optimal experience.* Harper Perennial, 1991.

[9] W. Schultz, P. Dayan, and P. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, pp. 1593–1599, 1997.

[10] P. Dayan and W. Belleine, "Reward, motivation and reinforcement learning," *Neuron*, vol. 36, pp. 285–298, 2002.

[11] S. Kakade and P. Dayan, "Dopamine: Generalization and bonuses," *Neural Networks*, vol. 15, pp. 549–559, 2002.

[12] J.-C. Horvitz, "Mesolimbocortical and nigrostriatal dopamine responses to salient non-reward events," *Neuroscience*, vol. 96, no. 4, pp. 651–656, 2000.

[13] M. Csikszentmihalyi, *Creativity-flow and the psychology of discovery and invention.* Harper perennial, 1996.

[14] J. Schmidhuber, "Curious model-building control systems," in *Proceeding International Joint Conference on Neural Networks*, vol. 2. Singapore: IEEE, 1991, pp. 1458–1463.

[15] S. Thrun, "Exploration in active learning," in *Handbook of Brain Science and Neural Networks*, M. Arbib, Ed. Cambridge, MA: MIT Press, 1995.

[16] J. Herrmann, K. Pawelzik, and T. Geisel, "Learning predicitve representations," *Neurocomputing*, vol. 32-33, pp. 785–791, 2000.

[17] J. Weng, "A theory for mentally developing robots," in *Second International Conference on Development and Learning.* IEEE Computer Society Press, 2002.

[18] X. Huang and J. Weng, "Novelty and reinforcement learning in the value system of developmental robots," in *Proceedings of the 2nd international workshop on Epigenetic Robotics : Modeling cognitive development in robotic systems*, C. Prince, Y. Demiris, Y. Marom, H. Kozima, and C. Balkenius, Eds. Lund University Cognitive Studies 94, 2002, pp. 47–55.

[19] F. Kaplan and P.-Y. Oudeyer, "Motivational principles for visual know-how development," in *Proceedings of the 3rd international workshop on Epigenetic Robotics : Modeling cognitive development in robotic systems*, C. Prince, L. Berthouze, H. Kozima, D. Bullock, G. Stojanov, and C. Balkenius, Eds. Lund University Cognitive Studies 101, 2003, pp. 73–80.

[20] J. Marshall, D. Blank, and L. Meeden, "An emergent framework for self-motivation in developmental robotics," in *Proceedings of the 3rd International Conference on Development and Learning (ICDL 2004)*, Salk Institute, San Diego, 2004.

[21] A. Barto, S. Singh, and N. Chentanez, "Intrinsically motivated learning of hierarchical collections of skills," in *Proceedings of the 3rd International*

Conference on Development and Learning (ICDL 2004), Salk Institute, San Diego, 2004.

[22] V. Fedorov, *Theory of Optimal Experiment.* New York, NY: Academic Press, 1972.

[23] D. Cohn, Z. Ghahramani, and M. Jordan, "Active learning with statistical models," *Journal of artificial intelligence research*, vol. 4, pp. 129–145, 1996.

[24] M. Hasenjager and H. Ritter, *Active learning in neural networks*, ser. Physica-Verlag Studies In Fuzziness And Soft Computing Series. Physica-Verlag GmbH, 2002, pp. 137–169.

[25] J. Denzler and C. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, no. 24, pp. 145–157, 2001.

[26] M. Plutowsky and H. White, "Selecting concise training sets from clean data," *IEEE Transactions on Neural Networks*, vol. 4, pp. 305–318, 1993.

[27] T. Watkin and A. Rau, "Selecting examples for perceptrons," *Journal of Physics A: Mathematical and General*, vol. 25, pp. 113–121, 1992.

[28] D. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, pp. 590–604, 1992.

[29] M. Belue, K. Bauer, and D. Ruck, "Selecting optimal experiments for multiple output multi-layer perceptrons," *Neural Computation*, vol. 9, pp. 161—183, 1997.

[30] G. Paas and J. Kindermann, "Bayesian query construction for neural network models," in *Advances in Neural Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7. MIT Press, 1995, pp. 443–450.

[31] K. O. M. Hasenjager, H. Ritter, *Active learning in self-organizing maps*. Elsevier, 1999, pp. 57–70.

[32] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, no. 2, pp. 201–221, 1994.

[33] J. Poland and A. Zell, "Different criteria for active learning in neural networks: A comparative study," in *Proceedings of the 10th European Symposium on Artificial Neural Networks*, M. Verleysen, Ed., 2002, pp. 119–124.

[34] J. Weng, "Developmental robotics: Theory and experiments," *International Journal of Humanoid Robotics*, vol. 1, no. 2, pp. 199–236, 2004.

[35] N. Roy and A. McCallum, "Towards optimal active learning through sampling estimation of error reduction," in *Proc. 18th Intl Conf. Machine Learning*, 2001.

[36] R. Collobert and S. Bengio, "Svmtorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2001.

[37] R. Sutton and A. Barto, *Reinforcement learning: an introduction.* Cambridge, MA.: MIT Press, 1998.

[38] C. Walkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[39] K. Kaneko and I. Tsuda, *Complex systems : chaos and beyond.* Springer, 2000.

[40] O. Sporns and T. Pegors, "Information-theoretical aspects of embodied artificial intelligence," in *Embodied artificial intelligence*, ser. LNAI 3139, F. Iida, R. Pfeifer, L. Steels, and Y. Kuniyoshi, Eds. Springer, 2003, pp. 74–85.

[41] J. Piaget, *The origins of intelligence in children.* New York, NY: Norton, 1952.

[42] O. Michel, "Webots: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[43] J. Rekimoto and Y. Ayatsuka, "Cybercode: designing augmented reality environments with visual tags," in *Proceedings of DARE 2000 on Designing augmented reality environments*, 2000, pp. 1–10.

[44] S. Schaal, C. Atkeson, and S. Vijayakumar, "Scalable techniques from non-parameteric statistics for real-time robot learning," *Applied Intelligence*, vol. 17, no. 1, pp. 49–60, 2002.

[45] E. Thelen and L. B. Smith, *A dynamic systems approach to the development of cognition and action.* Boston, MA, USA: MIT Press, 1994.

[46] R. D. Beer, "The dynamics of active categorical perception in an evolved model agent," *Adaptive Behavior*, vol. 11, no. 4, pp. 209–243, 2003.

[47] S. Nolfi and J. Tani, "Extracting regularities in space and time through a cascade of prediction networks," *Connection Science*, vol. 11, no. 2, pp. 129–152, 1999.

[48] M. Arbib, *The handbook of brain theory and neural networks.* Cambridge, MA: MIT press, 2003.

[49] M. Minsky, "A framework for representing knowledge," in *The psychology of computer vision*, P. Wiston, Ed. New York: Mc Graw Hill, 1975, pp. 211–277.

[50] R. Schank and R. Abelson, *Scripts, plans, goals and understanding: An inquiry into human knowledge structures.* Hillsdale, NJ.: Lawrence Erlbaum Associates, 1977.

[51] G. L. Drescher, *Made-up minds.* Cambridge, MA.: The MIT Press, 1991.

[52] R. Sutton, D. Precup, and S. Singh, "Between mdpss and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, pp. 181–211, 1999.

[53] K. Doya, K. Samejima, K. Katagiri, and M. Kawato, "Multiple model-based reinforcement learning," *Neural computation*, vol. 14, pp. 1347–1369, 2002.

[54] J. Tani and S. Nolfi, "Learning to perceive the world as articulated : An approach for hiearchical learning in sensory-motor systems," *Neural Network*, vol. 12, pp. 1131–1141, 1999.

[55] M. Tomasello, M. Carptenter, J. Call, T. Behne, and H. Moll, "Understanding and sharing intentions: the origins of cultural cognition," *Behavioral and Brain Sciences (in press)*, 2004.

[56] F. Dignum and R. Conte, "Intentional agents and goal formation," in *LNCS 1365: Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages.* London, UK: Springer-Verlag, 1997, pp. 231–243.

[57] F. Kaplan and V. Hafner, "The challenges of joint attention," *Interaction Studies*, vol. 7, no. 2, pp. 128–134, 2006.

[58] A. Robins, "Transfer in cognition," *connection science*, vol. 8, no. 2, pp. 185–204, 1996.

[59] G. Lakoff and M. Johnson, *Philosophy in the flesh: the embodied mind and its challenge to Western thought.* Basic Books, 1998.

[60] D. Gentner, K. Holyoak, and N. Kokinov, *The analogical mind: perspectives from cognitive science.* MIT Press, 2001.

[61] L. Pratt and B. Jennings, "A survey of connectionist network reuse through transfer," *Connection Science*, vol. 8, no. 2, pp. 163–184, 1996.

[62] J. Tani, M. Ito, and Y. Sugita, "Self-organization of distributedly represented multiple behavior schema in a mirror system," *Neural Networks*, vol. 17, pp. 1273–1289, 2004.

[63] F. Kaplan and P.-Y. Oudeyer, "The progress-drive hypothesis: an interpretation of early imitation," in *Models and mechanisms of imitation and social learning: Behavioural, social and communication dimensions*, K. Dautenhahn and C. Nehaniv, Eds. Cambridge University Press, to appear.

[64] L. Vygotsky, *Mind in society.* Harvard university press, 1978, the development of higher psychological processes.

[65] L. Steels, "The autotelic principle," in *Embodied Artificial Intelligence*, ser. Lecture Notes in AI, I. Fumiya, R. Pfeifer, L. Steels, and K. Kunyoshi, Eds. Berlin: Springer Verlag, 2004, vol. 3139, pp. 231–242.

[66] A. Meltzoff and A. Gopnick, "The role of imitation in understanding persons and developing a theory of mind," in *Understanding other minds*, H. T.-F. S. Baron-Cohen and D.Cohen, Eds. Oxford, England: Oxford University Press, 1993, pp. 335–366.

[67] C. Moore and V. Corkum, "Social understanding at the end of the first year of life," *Developmental Review*, vol. 14, pp. 349–372, 1994.

[68] P. Rochat, "Ego function of early imitation," in *The Imitative Mind : Development, Evolution and Brain Bases*, A. Melzoff and W. Prinz, Eds. Cambridge University Press, 2002.

[69] J. Baldwin, *Mental development in the child and the race.* New York: The Macmillan Company, 1925.

[70] H. Schaffer, "Early interactive development in studies of mother-infant interaction," in *Proceedings of Loch Lomonds Symposium.* New York: Academic Press, 1977, pp. 3–18.

[71] J. Piaget, *Play, dreams and imitation in childhood.* New York: Norton Press, 1962.

[72] J. Gibson, *The ecological approach to visual perception.* Lawrence Erlbaum Associates, 1986.

[73] Baillie, "Urbi: Towards a universal robotic low-level programming language," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS05*, 2005.
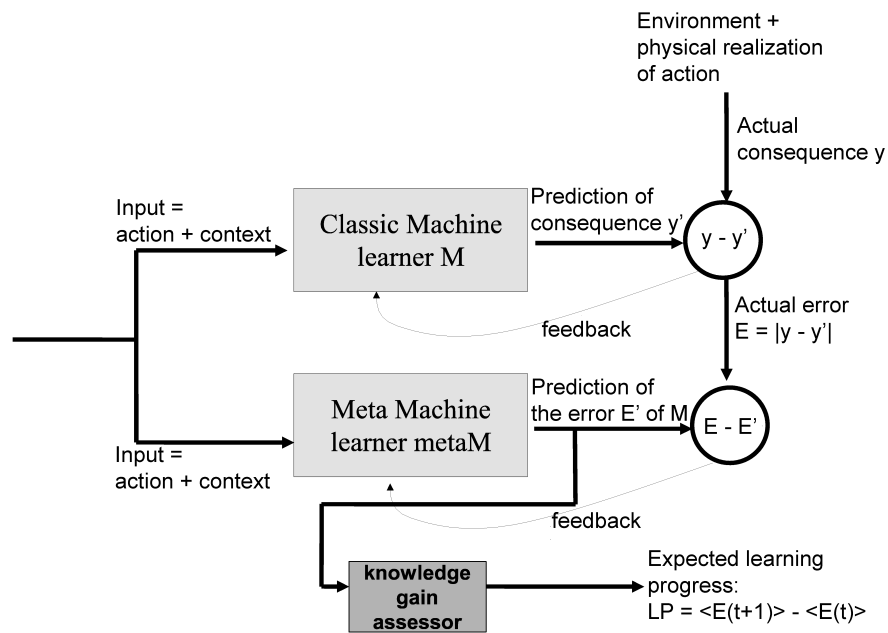
Figure 1: The architecture used in various models of group 2 and group 3: here there is a module **KGA** which monitors the derivative of the errors of prediction of **M**, which is the basis of an evaluation of learning progress. Some systems (group 2) evaluate the learning progress by measuring the decrease of the error rate of **M** in the close past, whatever the recent situations. Some other systems (group 3) evaluate the learning progress by measuring the decrease of the error rate of **M** in situations which are similar, but not necessarily close in time.
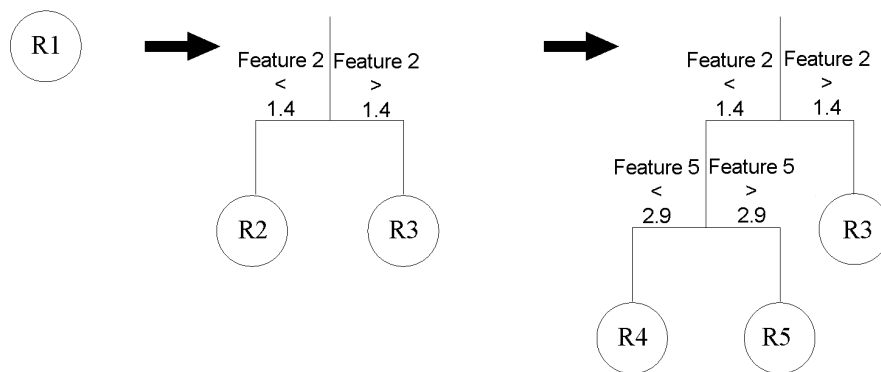
Figure 2: The sensorimotor space is iteratively and recursively split into sub-spaces, which we call "regions". Each region $\mathcal{R}_n$ is responsible for monitoring the evolution of the error rate in the anticipation of the consequences of the robot's actions if the associated contexts are covered by this region. This list of regional error rates is used for learning progress evaluation
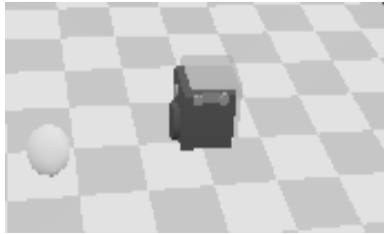
Figure 3: The robotic set-up: a two-wheeled robot moves in a room and there is also an intelligent toy (represented by a sphere) which moves according to the sounds that the robot produces. The robot perceives the distance between himself and the toy. The robot tries to predict this distance after performing a given action, which is a setting of (left wheel speed, right wheel speed, sound frequency). He chooses the actions for which it predicts its learning progress will be maximal.
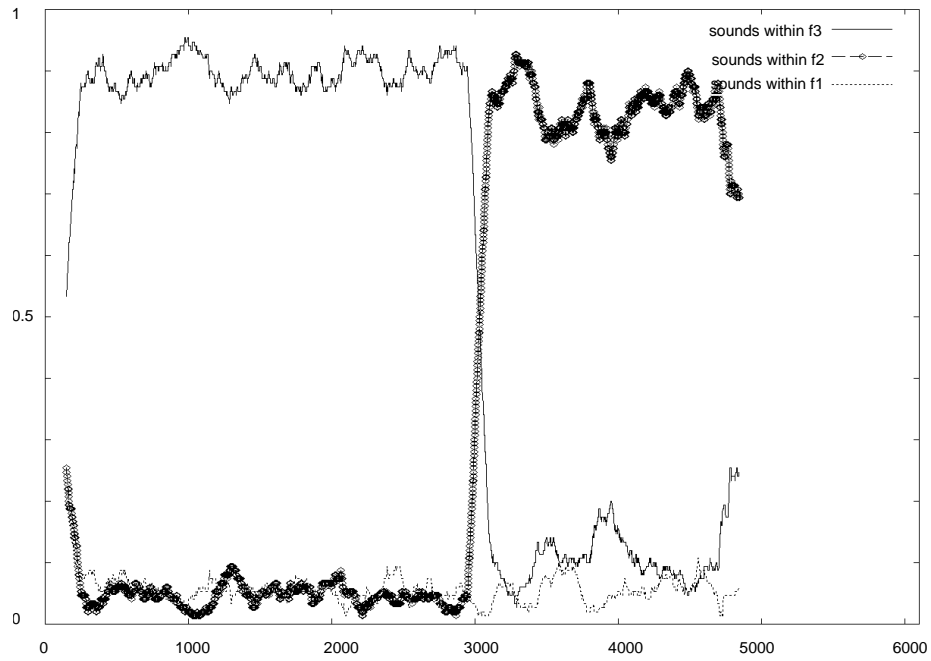
Figure 4: Evolution of the percentage of time spent in: 1) situations in which the emitted sounds have a frequency within f_3 (continuous line); 2) situations in which the emitted sounds have a frequency within f_2 (dotted line); 1) situations in which the emitted sounds have a frequency within f_1 (dashed line).
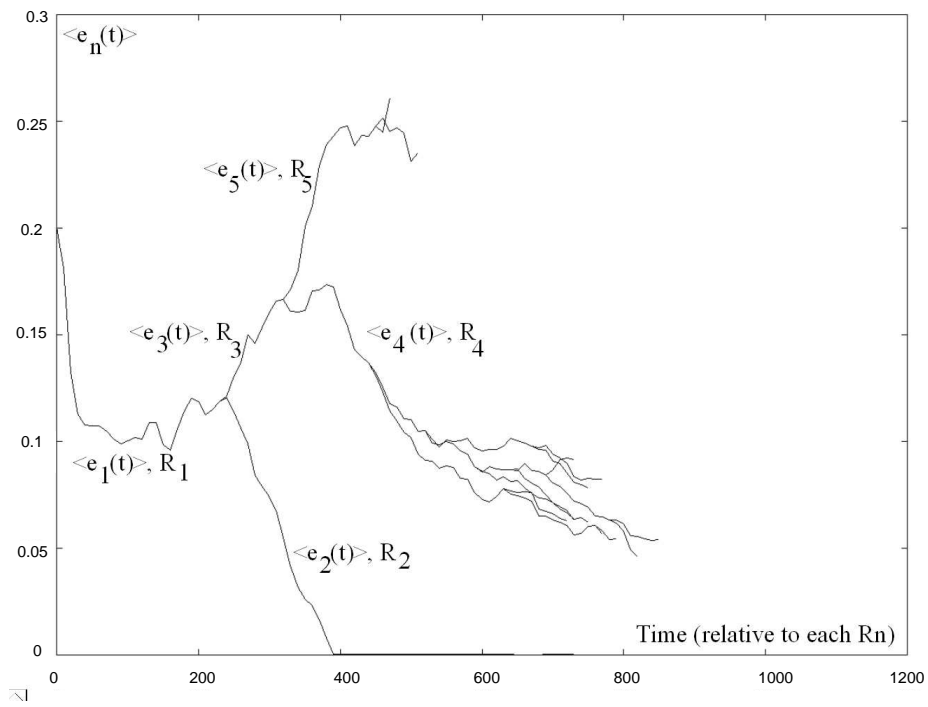
Figure 5: Evolution of the successive values of $< e_n(t) >$ for all the regions $\mathcal{R}_n$ constructed by the robot.
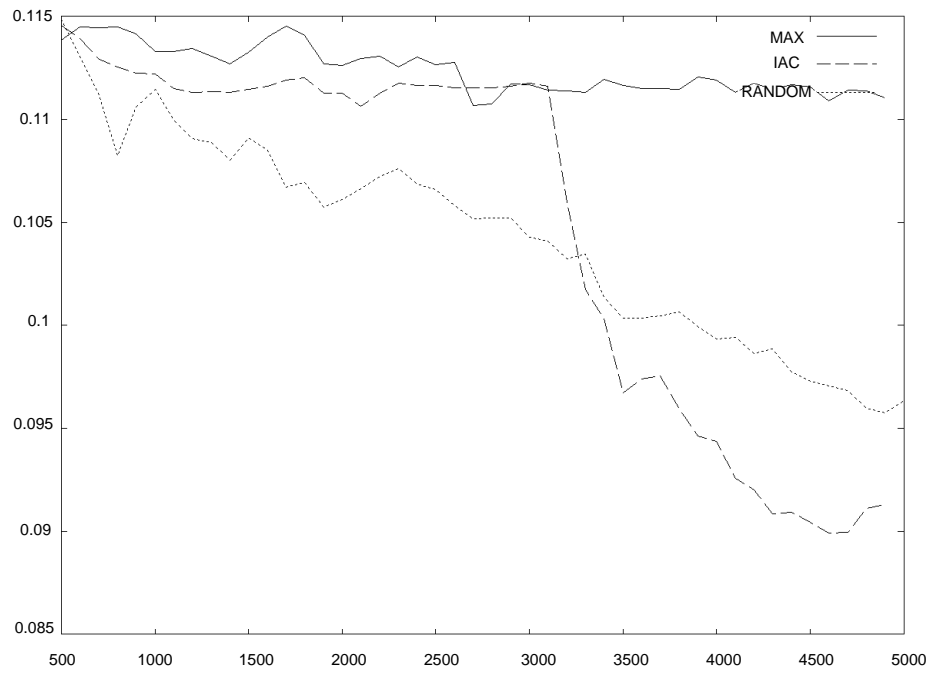
Figure 6: Evolution of the performance in generalization (mean squared prediction error) in situations in which the frequency of the emitted sound is within f_2, and respectively for the MAX algorithm (continuous line), the IAC algorithm (long dashes line) and the $RANDOM$ algorithm (small dashes lines). This allows to compare how much the robot has learnt of the interesting situations after a given number of performed actions, when it uses a given action selection algorithm.
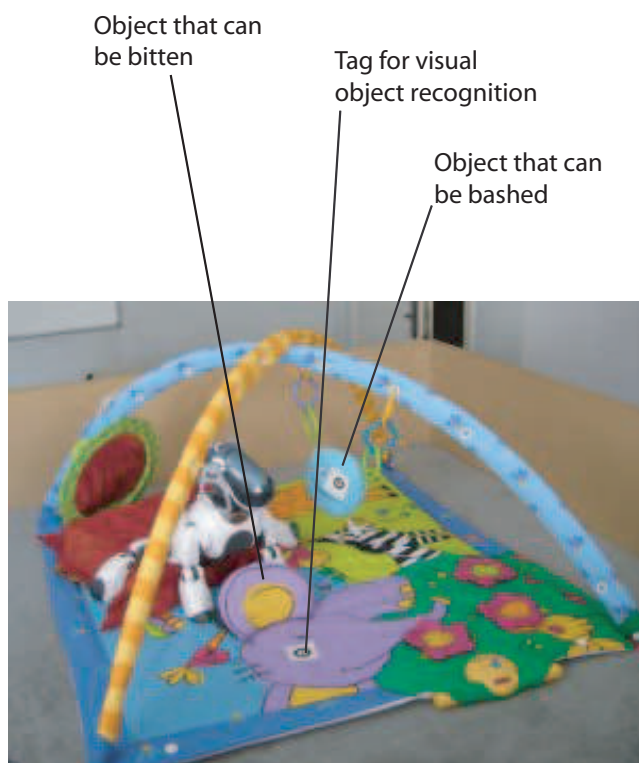
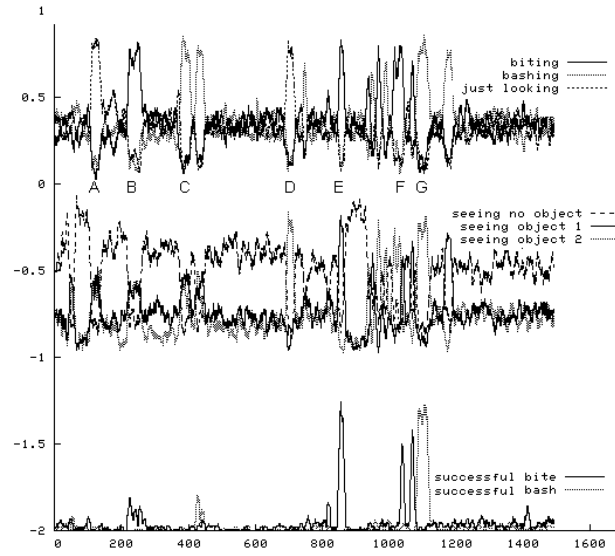Figure 7: The Playground Experiment set-up.

Figure 8: Curves describing a run of the Playground Experiment. Top 3: Frequencies for certain action types on windows 100 time steps wide. Mid 3: Frequencies of gaze direction towards certain objects in windows 200 time steps wide: "object 1" refers to the bitable object, and "object 2" refers to the bashable object. Bottom 3: Frequencies of successful bite ans successful bash in windows 200 time steps wide.
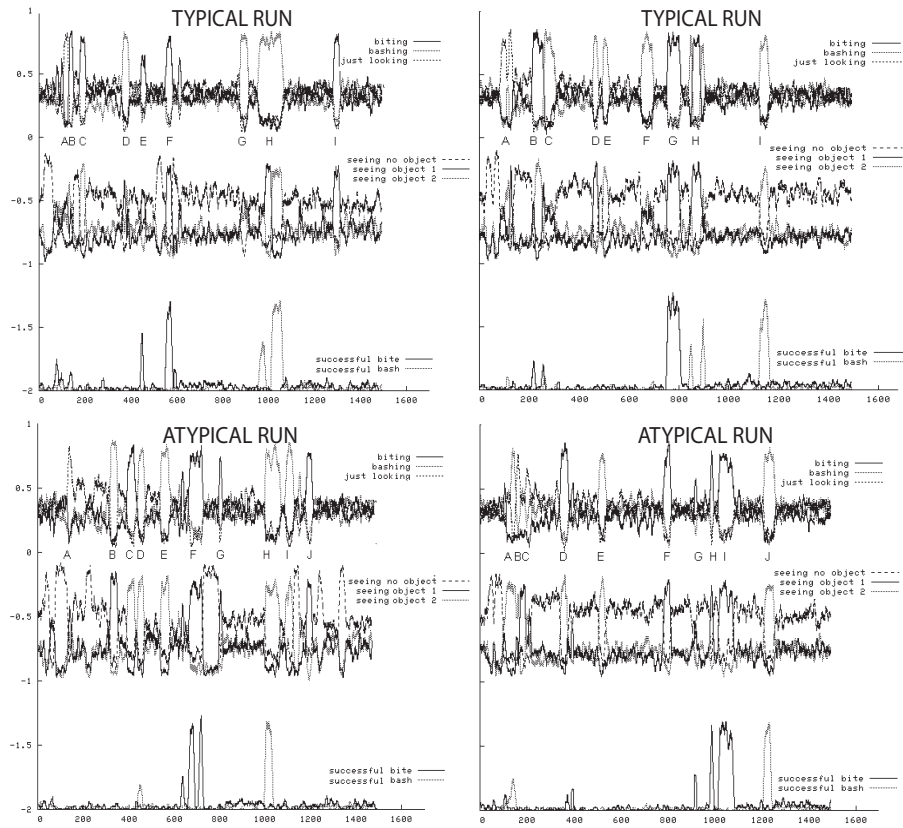
Figure 9: Various runs of the simulated experiments. In the top squares, we observe two typical developmental trajectories corresponding to the "complete scenario" described by measure 1. In the bottom curve, we observe rare but existing developmental trajectories.

Table 1: Statistical measures on the 200 simulation-based experiments.

Measures	Results
(1) number of peaks?	9.67
(2) complete scenario?	Yes: 34 %, **No: 66 %**
(3) near complete scenario?	**Yes: 53 %**, No: 47%
(4) non-affordant bite before affordant bite?	**Yes: 93 %**, No: 7 %
(5) non-affordant bash before affordant bash?	**Yes: 57 %**, No: 43 %
(6) period of systematic successful bite?	**Yes: 100 %**, No: 0 %
(7) period of systematic successful bash?	**Yes: 78 %**, No: 11 %
(8) bite before bash?	**Yes: 92 %**, No: 8 %
(9) successful bite before successful bash?	**Yes: 77 %**, No: 23 %