DOI: http://dx.doi.org/10.7551/978-0-262-31709-2-ch148

FIMO: Framework for Intrinsic Motivation

Fabien Hervouet and Eric Bourreau

LIRMM
Université Montpellier 2
fabien.hervouet@lirmm.fr, eric.bourreau@lirmm.fr

Abstract

This paper aims to introduce a flexible framework called *FIMO* dedicated to intrinsic motivation in developmental robotics. With this framework we want to offer a common way of implementing, testing and analyzing mechanisms related to intrinsically motivated algorithms. It may be seen as a generic complex open source framework for online exploration and structuring of learning spaces. We hereby lay both theoretical and practical foundations of our framework to encourage future experimental studies.

Introduction

A commonly shared objective, in the developmental robotics community, as its name indicates, is insisting on the developmental part. It means that the most important aspect of any research leaded in this field lies in the path that leads to staged growth of learning. It may be about practical competences, conceptual comprehension or whatever a living organism with physical capabilities can practice. This implies that our goal is all about defining a model of developmental mechanisms, allowing a full open-ended incremental learning in an autonomous and interactive way, compliant and resilient with a real environment. The challenge is to propose a compromise between conciseness of the model and complexity it can generate as soon it is instantiated. But for an agent, developing competences is not that easy. It has to start with some coarse grained capabilities which are going to be refined throughout the developmental phase and life in general. This upgrowth is characterized by a double brain/body maturity, but also by the acquisition of sensorimotor experiences supporting learning. Inside the developmental robotics community, this is currently realized by inspecting multiple aspects of development. First the individual autonomous mental development which insists on finding mechanisms for an interactive system to develop itself solely using its very low-level sensory inputs and acting through its low-level motor outputs. The main second path for exploring development is the social learning. In this paper, we are interested in the first view.

Precisely, our working frame is mechanisms that can play the role of heuristics in order to focus and control the exploration of the potentially huge sensorimotor space of a situated and fully embodied agent. We believe that the goal of any developmental robotics algorithm is to design a control loop that – if executed on robot with physical capabilities – should reveal its own morphology affordance within its environment, through an unsupervised process. Moreover we also believe, as we already said, that the key challenge is to identify and implement low-level mechanisms that allow a long-term development as a scaffolding of capabilities. The more low-level these mechanisms are, the more the system can be considered as relevant. In the case of bio-inspired developmental robotics, we are interested in sensorimotor learning through the open-ended exploration of high-dimensional and complex bodies. This means we have to inspect and design scalable task-independent mechanims that may involve the robot in a self autonomous skill practice. The main idea remains that the huge sensorimotor space can be divided into subspaces in order to facilitate the learning of the full space using intrinsic heuristics of space exploration. Another way to explain our view is to say that we are aiming at transfering the traditional cognitive modeling biases towards the natural limits and constraints imposed by the sensorimotor grounding embodiment.

In our case, we here underline the need for a framework for intrinsic motivation as we introduce it, with the capability to make parametric studies and introduce new bio-inspired ideas from state of the art neuroscience or developmental psychology research. We insist on the fact we propose in our framework *FIMO* facilitations for future improvements for some parts of the general intrinsically motivated algorithm it implements. In the rest of the article, we first propose a motivational background of the specific research field of ours. Then we present the practical foundations of this framework called *FIMO*. Then we zoom on the theoretical model of our view of intrinsic motivations. We conclude this paper by recalling the need for such a framework in our community, and draw some interesting application perspectives.

Motivational Background

The question to ask when we get to implement some mechanisms like motivation in artificial agents, is about the origin of the notion of motivation. In this case, psychology has studied motivation a lot. The very first background of works about intrinsic motivation is the self-determination theory, which is the result of psychological research leaded by Deci and Ryan (1985) from the human point of view. This theory says that any individual has innate tendencies towards personal growth and vitality that are either satisfied in relation to their immediate environment. The theory also explains that there are three satisfactory needs that actors seek to satisfy: competence, relatedness and autonomy. Innate tendencies are enacted when these needs are fully satisfied. We will see that these founding psychological works have been at the base for transfering the notion of motivation *in silico*.

Historically, Schmidhuber (1991a) was the first to introduce research results about the importance of what he called *artificial curiosity*, in the sense of applying typical *human* curiosity to *artificial* machines. He argues that his research work has been driven by the simple idea that an optimal adapted and motivated agent is nothing but an agent trying to improve its compressed model of the world. He said that an agent with a prediction module, allowing the lossless compression of data by finding regularities in the world, is driven toward an optimal improvement. Schmidhuber (1991b) proposed to measure the learning progress of an agent by comparing difference of prediction for a same situation before and after the reality feedback. He therefore introduced the idea that an optimal curious agent's interest lies in the narrow corridor between what is simply too compressible and therefore uninteresting and boring, and what is not compressible at all because of a lack of regularity making it too complicated to learn.

Barto et al. (2004) have extended their own work in reinforcement learning (Sutton and Barto, 1998) with the notion of intrinsic motivation. They introduce an elaboration of the existing reinforcement learning framework that "encompasses the autonomous development of skill hierarchies through intrinsically motivated reinforcement learning". Their model advocates the creation and use by an agent of structures of instrinsic generic rewards allowing adapted behavioral learning. They do not consider in any way external rewards.

Soon after the birth of the developmental robotics community (see Weng et al., 2001), Oudeyer and Kaplan (2004) introduced the *IAC* (*Intelligent Adaptive Curiosity*) algorithm. It explicitly refers to Schmidhuber's adaptive curiosity and must be linked with another paper published the same year by Steels (2004). The proposed mechanism is anchored at a sensorimotor level and allows low-level action selection in the high-dimensional sensorimotor space for a robot. This algorithm postulates that one way to provide autonomy to a robot is to let it make its proper action choices, based on its experience, in order to maximize its learning. With this architecture an embodied agent is going to experiment grounded sensorimotor coordinations in order to learn the effects of its actions thanks to a unique action selection mechanism that tends to choose actions that improve prediction quality.

Baranes and Oudeyer (2010) then proposed an evolution to the original *IAC* algorithm called *SAGG-RIAC* which is of interest to us because of the competence acquisition paradigm it explores. The global principle remains the same except that this time, the agent has to choose sensory regions where it wants to return to, instead of choosing sensorimotor regions where it comes from. Practically, the *SAGG-RIAC* algorithm is based on alternating reaching phases (i.e. reaching a goal in what they call an operational space) and local exploration phases (i.e. improving the world comprehension toward the goal). The purpose of reaching phases is to test the reliability of the forward motor model while the purpose of exploration phases is to improve the inverse model of the system in the close vicinity of the current state. Exploration phases are triggered when the reliability of the local controller is too low. In the following section we explain some improvements to this algorithm we would like to introduce and experiment.

This research must also be linked with some other works by Blank et al. (2005) we fully agree with, where they argue that any intrinsic developmental algorithm has to be based on a recursive model that produces complex behavior and that it should rely on three concepts: abstraction, anticipation and intrinsic motivation.

General Architecture of *FIMO*

FIMO[1] has the goal of bringing up a brand new open, flexible and extensible framework for research in the developmental robotics field working on agents driven by intrinsic motivations. We propose with this architecture, a solid and well-thought way of experimenting new ideas for the intrinsic motivation interested community. This section is dedicated to the presentation of *FIMO*, either the way it is architectured, its pythonic foundations, its software architecture and workflow, and the default provided environments.

Python Foundations

First of all, it is important to explain the Python foundations of this framework. We based our development on well documented, community supported and very powerful libraries such as *SciPy, NumPy, matplotlib* which are dedicated to scientific data processing. The first one Jones et al. (2001) is open-source software for mathematics, science, and engineering. It depends on the second one, which provides con-

[1]The Python open source code of the framework is available at `https://info-depot.lirmm.fr/republic/fimo` (public cloning repository) released under GNU GPLv3 license.

venient and fast *N-dimensional array manipulation*, with efficient numerical routines. The visualization module relies for its part on the third one Hunter (2007). Our point is that the techniques used in this framework have been already deeply tested and used by a lot of other researchers, and that they have proved their efficiency and their reliability.

Software architecture

The global workflow of our framework *FIMO* is rather common and intuitive. Typically, the framework features an environment that describes in a formal way the being explored sensorimotor space, which is mandatory for bootstrapping the agent's living loop. Once the environment is chosen, we must also provide the agent with parameters and metrics – which may have a significant influence on its developmental trajectory – depending on the choices made at the instantiation time. Then, at the end of a run, i.e. when the program has tried to reach the certain amount of goals you asked it to perform, multiple data are dumped into log files. Typically, it includes the evolution of space partitioning into subregions, acquired sensorimotor raw data, interests of selected regions where objectives were generated and test results. Finally, another powerful advantage of the framework also lies in the visualization module made possible thanks to the logging process. This possibility to observe the result of any run instance makes it easier to understand and validate the configured parameters, metrics and other implemented ideas. We will illustrate the rest of the paper with some visualizations *FIMO* can offer.

Environments

An environment as we defined and formalized it within our framework is all about the agent's acting possibilities. We mean that an environment should be seen as a complex interface between the sensory (or operational) space to explore, the motor space allowed for this exploration, and the combined result of these two. An environment must provide the sensory dimensions or operational dimensions, the motor dimensions, a starting state or rest position. It must also specify a function computing a new state given a current state and an action to execute, and a function that generates reachable coordinates in the sensory / operational space to be used for the examination. As a bias, you can predefine a region partitioning for the agent to start with, in order for instance to help it bootstrapping. The way we implemented the connection between environments and the main loop must allow to easily connect to some existing 3D simulated environments or to any physical robots following the same guidelines. Table 1 provides a summary of proposed simulated environments in *FIMO* and present their sensory and motor dimensions.

Wheeled vehicle The first provided environment is a very simple wheeled vehicle in an arena. The first version is only two dimensional (1D sensory space and 1D motor space).

Environments	a	b
One-wheeled vehicle (OWV) 1D fully reachable	1	1
OWV 1D not fully reachable	1	1
OWV 1D bump requiring sufficient inertia to pass	1	1
Two-wheeled vehicle (TWV) 2D squared area	2	2
TWV 2D squared area with obstacles	2	2
TWV 2D triangled area	2	2
Robotic arm (RA) with one joint	2	1
RA with two joints	2	2
RA with three joints	2	3
RA with fifteen joints	2	15

Table 1: Summary of the variations of the main existing environments in *FIMO* presenting their distinctive feature(s), the size of their operational (a) and motor (b) spaces.

The second version is four dimensional (two sensory space, 2D motor space). Generally the shape of the arena is a square, but it could be different or contains obstacles. In the first version, the vehicle evolves in a 1D space and has a sole coordinate (x), that is the distance to the front wall. In the second version, the vehicle evolves in a 2D space and has two coordinates $(x; y)$. It can move by performing an action (Δ_x, Δ_y) representing a shift in the plane.

Robotic Arm We propose a second environment typically used – in particular in Baranes and Oudeyer (2010, 2013) – to test intrinsic motivation algorithms: a robotic arm (cf. figure 1[2]). The environment provided is a generic robotic arm that you can easily instantiate by defining the number of joints/limbs you want, and their respective length (or the unique length if ever).

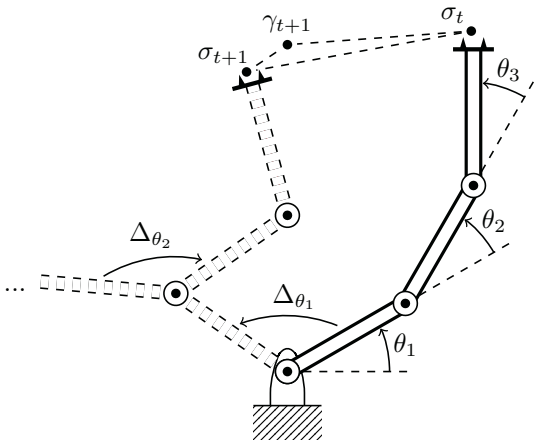


Figure 1: Schematic successive positions of a 3 joints robotic arm environment at time t and $t + 1$.

[2]Here, each limb has the same length and θ_i represent angles of each joint, relatively to θ_{i-1}. The performed action that moves the end-effector from σ_t to σ_{t+1} coordinates in the operational space by trying to reach γ_{t+1} was $\alpha = (\Delta_{\theta_1} = +115, \Delta_{\theta_2} = -140, \Delta_{\theta_3} = +40)$.

The transition from end effector position to another one through a given action is computed using standard methods for forward position kinematics. An action is a vector of positive or negative Δ for each joint $(\Delta_{\theta_1}, \Delta_{\theta_2}, \Delta_{\theta_3})$. It means the agent can only use relative moves from one $(H_{x,t}, H_{y,t})$ to the next $(H_{x,t+1}, H_{y,t+1})$ position.

Theoretical Model

In this section we explain the theoretical model behind of *FIMO*. Thus, we present the main algorithm and every single parameters or metrics that the experimenter must use as such or improve, extend or change. Nevertheless, what we want to emphasize is that this algorithm needs to be configured by many parameters and metrics to be specified by the experimenter that can strongly influence the final behavior of the system and its evaluation.

The living loop algorithm as we call it, proposes a kind of reinforcement learning method, but it must be seen as an empty shell that must be completed with the right parameters to be able to observe the best behavior (see algorithm 1). Indeed, in the same way that one must specify an environment, the configuration parameters and the choice of metrics are essential and grouped in a *config.py* file. The interaction of the body/environment and the intrinsically motivated algorithm is something very delicate which requires the attention of the experimenter. One must see this guided exploration of sensorimotor space as a *complex system* as it is composed with multiple interconnected parts (different parameters and metrics to tune) which as a whole exhibit behavior not obviously predictable from the individual properties. The interaction of these parameters put together in the living loop algorithm may exhibit emergent properties. Moreover we believe that it may be possible to define some precise optimized setting for a specific environment.

In the rest of the section we present some of the general part of the framework (main algorithm, default regions structure and implemented learning method) as well as the criticals part that may be overriden with improvements (competence and interest measures, the way memory may be restructured). These parameters should be deeply studied in order to fit the need for a specific morphology, because we believe that the embodiment plays a major role in order to help determining fully appropriate, responsive, efficient and developmental settings.

Thus this is all about trying to find a compromise to encourage and facilitate effective and efficient exploration of space, without losing time in not interesting areas, while allowing rapid improvement for test result. Among the default implementation choices we present here, some have already been presented in Hervouet and Bourreau (2012).

Living loop algorithm

Although we keep the overall operation of the original motivational living algorithm SAGG-RIAC Baranes and Oudeyer (2010), we extended the frame. The global

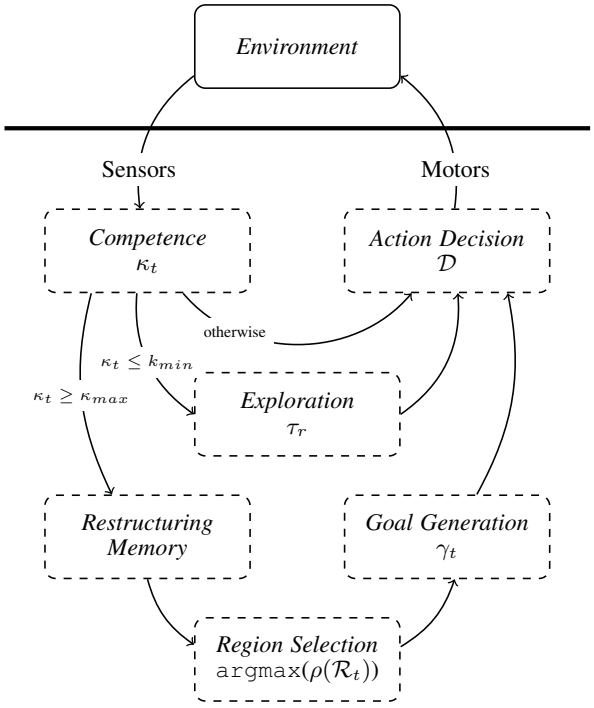


Figure 2: General algorithmic architecture of the intrinsically motivated living loop implemented in *FIMO*.

algorithmic architecture is presented in figure 2 and its implementation is presented in algorithm 1. With this implementation, we want to underline the flexibility of certain specific parts of the algorithm. We mean that the sequence of the algorithm is nothing but a generic system that aims at *exploring and structuring a sensory / operational space with partial data for sensorimotor learning*. Indeed we have to be aware of the strong influence of metrics and parameters that we may use to obtain more compliant or efficient behavior. This is the reason why we decided to emphasize the critical parts in the algorithm in order to be more illustrative. Each of them is studied in the rest of this section.

In a formal way, our architecture makes a distinction between *raw data* ξ_r accumulated during *exploration phases* (second loop in fig. 2) and more *structured data* ξ_g accumulated during *exploitation phases* (third loop in fig. 2). An exploration phase is triggered each time the agent is considered as incompetent and consist in accumulating τ_r raw data about the consequences of actions the agent executes. In practice, exploration data represent a forward model of the body in the world. Exploitation phases consist in generating a goal and attempting to reach it in τ_g times. Exploitation data represent the particular historical and motivational coupling between the agent and its environment, in the sense of Varela et al. (1991), i.e. the sensory configuration he has set itself for its goals that it tried to achieve by itself with varying degrees of success, according to the increase of its competence in relation to these goals.

As it is implicitely formalized, time is considered in the algorithm: σ_t always represents the agent's current state, which changes every time the agent performs an action α_t using the *execute(α)* method.

Algorithm 1: Intrinsically Motivated Living Loop

input: σ experienced states;
$\mathcal{R}$ set of existing regions;
γ set of self generated goal;
ρ interest measure;
κ competence measure;
$\mathcal{D}$ action decision method;
ξ_r, ξ_g respectively raw and goal experiments;
τ_r, τ_g resp. exploration and reaching trials;

1 **while** *True* **do**
2 $\quad \sigma_{start} \leftarrow \sigma_t$
3 $\quad \mathcal{R}_t \leftarrow \texttt{argmax}(\rho(\mathcal{R}_i)\ \forall \mathcal{R}_i \in \mathcal{R})$ (see section IM)
4 $\quad \gamma_t \leftarrow \texttt{randomGoal}(\mathcal{R}_t)$
5 $\quad \mathcal{A} \leftarrow \emptyset$
6 $\quad$ **repeat**
7 $\quad\quad \alpha_i \leftarrow \mathcal{D}(\xi_r, \sigma_t, \gamma_t)$ (see section AD)
8 $\quad\quad \mathcal{A} \leftarrow \mathcal{A} \cup \alpha_i$
9 $\quad\quad \texttt{execute}(\alpha_i)$
10 $\quad\quad \kappa_t \leftarrow \kappa(\sigma_{start}, \gamma_t, \sigma_t)$ (see section CM)
11 $\quad\quad \xi_r \leftarrow \xi_r \cup (\sigma_{t-1}, \alpha_i, \sigma_t)$
12 $\quad\quad$ **if** $\kappa_t \leq \kappa_{min}$ **then**
13 $\quad\quad\quad \xi_{g,\mathcal{R}_{\sigma_t}} \leftarrow \xi_{g,\mathcal{R}_{\sigma_t}} \cup (\sigma_{t-1}, \sigma_t, \{\alpha_i\}, \sigma_t, 0)$
14 $\quad\quad\quad$ **repeat**
15 $\quad\quad\quad\quad \sigma_t \leftarrow \sigma_t - 1$
16 $\quad\quad\quad\quad \alpha_j \leftarrow \texttt{randomAction}()$
17 $\quad\quad\quad\quad \texttt{execute}(\alpha_j)$
18 $\quad\quad\quad\quad \xi_r \leftarrow \xi_r \cup (\sigma_{t-1}, \alpha_j, \sigma_t)$
19 $\quad\quad\quad$ **until** τ_r *trials not exceeded*
20 $\quad\quad$ **end**
21 $\quad$ **until** $\kappa_t \leq \kappa_{max}$ *or* $|\mathcal{A}| \leq \tau_g$
22 $\quad \xi_{g,\mathcal{R}_t} \leftarrow \xi_{g,\mathcal{R}_t} \cup (\sigma_{start}, \gamma_t, \mathcal{A}, \sigma_t, \kappa_t)$
23 $\quad \texttt{restructuringMemory()}$ (see section RM)
24 **end**

Interest Measure (IM)

The measure of interest ρ qualifies the dynamic interest of a region. It is used to compute the most interesting region, the one with maximum ρ value (line 3), the agent is going to self-generate a goal in. The default one proposed in *FIMO* is based on the one introduced by Baranes and Oudeyer in Baranes and Oudeyer (2010) with one major difference. It is computed as follows:

$$\rho(\mathcal{R}_i) = \texttt{learningProg}(\mathcal{R}_i) + \texttt{diversification}(\mathcal{R}_i)$$

The learning progress is computed using experiments from exploitation phases (ξ_g for goal experiments) which are of the form:

$$\xi_{g,t} = \ (\sigma_t, \gamma_t, \mathcal{A}, \sigma_{t+i}, \kappa_t)$$

with σ_t the current sensory configuration state, γ_t the chosen goal to be reached, $\mathcal{A}$ the motor configurations successively performed to achieve the goal with $|\mathcal{A}| = i$, σ_{t+i} the new current state after execution of actions and the competence κ_t (explained in detail in a following section). The learning progress computation can be seen as a derivative of competences. Let κ_j be the competence of the j^{th} experiment stored in memory, and $|\mathcal{R}_i|$ the number of experiments in a region.

$$\texttt{learningProg}(\mathcal{R}_i) = \frac{|\sum_{j=0}^{|\mathcal{R}_i|/2} \kappa_j - \sum_{j=|\mathcal{R}_i|/2}^{|\mathcal{R}_i|} \kappa_j|}{|\mathcal{R}_i|}$$

We chose to directly incorporate a UCT based Kocsis and Szepesvári (2006) diversification measure which takes into account in an incremental way the number of experiments conducted in the current region relative to the total number of experiments. It means that this mechanism will gently wake up regions whose direct learning progress is decreasing but for which it would be wise to generate a goal in just to make sure everything has been already either understood or misunderstood. Let n and n_i be respectively the total number of experiments and the number of experiments in the current region. Let c be a constant that allows to normalize the result of diversification measure.

$$\texttt{diversification}(\mathcal{R}_i) = c \times \sqrt{\frac{\ln n}{n_i}}$$

In summary, the interest measure ρ is composed by the addition of two dynamic but separate measures. The first one computes the learning progress in a given region, while and the second one tries to uniformly balance against the natural excessive regional intensification of the first one. This separation of the intensification measure and the diversification measure is of real interest for future improvements, because we will only have to deal with precisely defining the first one without taking into account the uniformisation process preventing overspecialization.

Action Decision (AD)

The default decision method we propose is a simple algorithm that chooses the next action to perform in order to reach a goal driven through experience (line 7). We propose to compute action towards a goal using *k-nearest-neighbour* experiments chosen among previously acquired explorative experiments. These experiments (ξ_r for raw experiments) are of the form:

$$\xi_{r,t} = \ (\sigma_t, \alpha_t, \sigma_{t+1})$$

with σ_t the current sensory configuration state, α the motor configuration determined at time t and finally σ_{t+1} the new current state after execution of action α. These experiments must maximize two criteria: the initial and final states should be as close as possible respectively to current and goal states. The strategy then consists in generating a mean

action with respect to actions performed in these filtered experiments (with $|K| = k$).

$$\mathcal{D}(\xi_r, \sigma_t, \gamma_t) = \frac{\sum_{u=1}^{|K|} \alpha_u}{k} \text{ where}$$

$$K = \{\xi_{r,u} \in \xi_r : \ min(|\sigma_t - \sigma_u| + |\gamma_t - \sigma_{u+1}|)\}$$

Competence Measure (CM)

Competence κ_t ranges in $[\kappa_{min}; 0]$, as it is computed in the original algorithm, with κ_{max} typically equal to -1. The default implemented measure sums distances for all sensors in S between the final sensory position minus the goal sensory position, relative to the start position minus the goal position. It means that we compute the shift between where the agent comes from, where it should have arrived and where it has finally arrived.

$$\kappa(\sigma_s, \gamma_t, \sigma_f) = max(-\frac{|\sigma_f - \gamma_t|}{|\sigma_s - \gamma_t|}, \kappa_{min})$$

with σ_s, γ_t and σ_f respectively start, goal and final state κ_{min} the minimal competence (or maximal incompetence) and κ_{max} the very competent threshold. The 3D visualization of the competence value of the robotic arm evolving in a 2D space is presented in figure 3. We can observe the reachable area at the center.

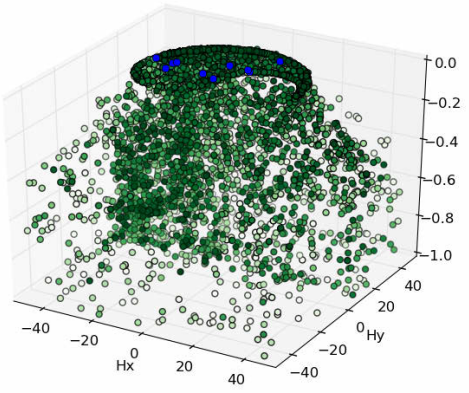


Figure 3: 3D visualization of generated goals for an instance run of the robotic arm in the two dimensional operational space. The closer to 0 on the third axis the goals, the more competent the agent was to reach them. The greener the goals, the more recently they have been reached. Blue points represent goals for assessment.

Restructuring Memory (RM)

A typical developmental and incremental process pushes the robot to start its learning from scratch. It can only make non-optimal decisions solely from previously acquired data. This implies that the robot acts very strangely at the begining because it does not hold enough information about the world it develops in. The figure 4 presents the result of a run instance with original operational space split into subregions of interest through time[3]. We can observe the naturally delimited reachable space because we add perfectly reached goals from where we arrive at the end of a reaching attempt. Moreover we can observe the reinforcement of generated goals in the vicinity of these natural embodiment limits. This shows that the agent seems to find interesting areas located near the limits it can reach, which must be considered as a good behavior.

Therefore, pushing the robot to split its sensorimotor space allows it to overcome the lack of information at the beginning of the developmental living process. This is this particular splitting condition that makes the strength of this approach because it allows *the isolation of coherent experiments*. This coherence may depend on the nature of the measure that determines the best split in a sensorimotor region. The default coherence measure implemented in *FIMO* is related to the notion of learning progress[4], i.e. the derivative of learning.

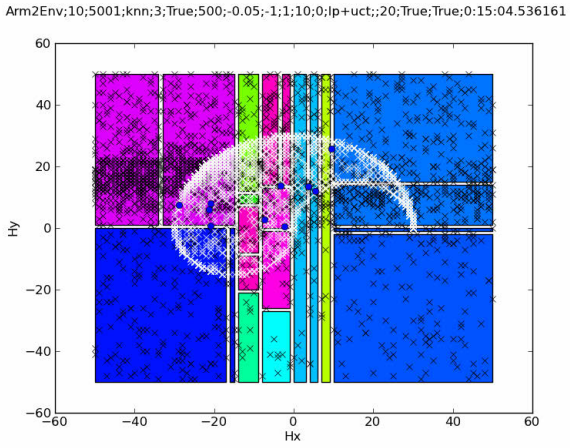


Figure 4: Visualization of generated experienced goals and regions containing them in the split operational space for a robotic arm environment.

Thus the splitting condition needs to take into account knowledge and experience accumulated. Otherwise the agent will always tend to split again and again its space, which will probably be pointless. We wanted to propose a general way to be able to implement a kind of dual restructuring measure. Because it may be very interesting to

[3]The shape of the reachable area is not a full circle because of the limitation we added for each joint's absolute angle θ_i to range between $[0; 180]$.

[4]But it could be related to any other idea that we could think of progress: better structuring memory, novelty, compression, etc. This is the reason why we should facilitate the implementation of new ideas by emphasizing replaceable metrics and parameters.

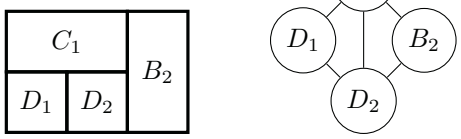
be able to elaborate a measure that would decide, depending on accumulated experience, not only how to split, but more how to reorganize the memory. This vision totally fits a developmental process. Thus we chose to combine these two mechanisms by proposing by default a dual measure for both splitting and merging. It tries to maximize the absolute value of the difference between the learning progress in the two subregions relative to the current learning progress in the mother region.

$$\mu(\mathcal{R}_1, \mathcal{R}_2, R) = \frac{|LP(\mathcal{R}_1) - LP(\mathcal{R}_2)|}{LP(R)}$$

It means that in the splitting case we will split the current region R if it contains two subregions $\mathcal{R}_1$ and $\mathcal{R}_2$ exhibiting better learning progress, relatively to the current learning progress in the region R. On the other hand in the merging case, we will merge two regions that have been split possibly a long time ago when the agent didn't hold enough information: the current region $\mathcal{R}_1$ should be merged with another region $\mathcal{R}_2$ into a sole region R if the learning progress of R exhibits a better evolution than $\mathcal{R}_1$ and $\mathcal{R}_2$.

Regions as a graph structure

In order to propose more genericity for future implementations we chose to upgrade the tree structure of regions used since *IAC* Oudeyer and Kaplan (2004). We decided to introduce a graph structure of regions (cf. figure 5) because we considered that the tree representation was a limitation. We believe that, by its more scalable and flexible nature, the graph structure facilitates from an operational point of view potential reorganization as we will discuss in the following section. In particular it may allow the creation of non-convex regions by merging already existing region whether they are physically adjacent or not.



(a) Typical final result of the space splitting process after a run instance.

(b) Graph structure of the space splitting (edges represent adjacency between two regions).

Figure 5: Illustration of the graph structure for regions. On the left it represents the result of a 2-dimensional sensory space splitting process, while on the right the way this is handled in memory.

So, how to evaluate?

To propose a framework laying the foundations to encourage the community to easily implement and test new ideas is one thing, to propose a way to evaluate them is another. Indeed setting up such a framework obviously also involves setting up an experimental validation process as a common foundation for evaluation, which is not a trivial thing. As pointed out by Meeden and Blank (2006) a few years ago, the evaluation question is fundamental, especially in the frame of autonomous developmental robotics.

We generally make a distinction between the formative and the summative assessments in the literature. A quotation attributed to Robert Stakes in Scriven (1991) tends to explain the difference between them: *"When the cook tastes the soup, that's formative; when the guests taste the soup, that's summative"*.

Formative assessment represents the global operation of the intrinsically motivated process proposed living loop algorithm. Indeed, the main role of formative assessment lies in its subjective regulatory function of the learning process within the system, where the learner must be able to measure progress made and progress to be made.

Summative assessment must be seen as a more common and accepted way permitting to test learning acquisition from outside of the system, by certificating whether knowledge has been assimilated. This means we must provide in *FIMO* an external way, considered as objective as possible, to measure the relevance and importance of potential improvements as explained in Hervouet (2013). Practically in *FIMO* we implemented a measure that circumbscribes the evaluation of the evolution of the competence through time. The idea is to select a set of arbitrary uniform reachable goals and compute a mean incompetence for reaching them.

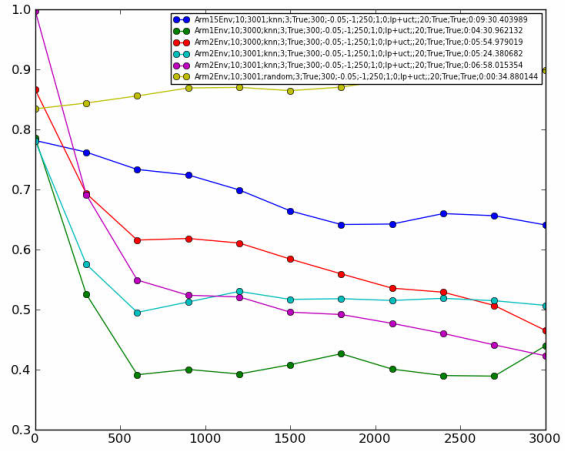


Figure 6: Evolution of the mean incompetence for a given set of reachable goals for different environment and parameters (1 means bad competence while 0 means very competent).

As an illustration, we propose on figure 6, a comparison of assessment scores for a set of different settings under different environments. From bottom: (1) 1-joint robotic arm; (2,3,4) three instanciations with the same settings of a 2-

joints robotic arm; (5) 15-joints robotic arm; (6) 2-joints robotic arm but with random action (baseline).

Conclusions & Perspectives

In this paper we argue that in the developmental robotics community, and more specifically in the subcommunity interested in intrinsically motivated robotics, there are multiple ways to propose some improvements. With the proposed framework *FIMO*, we clearly made an attempt, starting from an existing approach, to extend the frame. It is absolutely not the purpose of this article to introduce some fascinating new metrics or define more precisely some of the parameters. We are not saying we have found the best something or a better somewhat. In contrast, we believe that this will be the purpose with forthcoming publications, thanks to the implementation of our framework and the facilitations it brings up for the evaluation and comparison of future improvements. Notwithstanding, there are various parameters waiting to be tuned. However, the fact remains that, to illustrate this purpose, we proposed some default parameters, metrics or even other surroundings learning features, but they are presented as an indication because we were confronted to the need to set up the framework with implementation choices.

We consider *FIMO* as a necessary formal step toward an open future of intrinsic motivations work because it will help future contributions and improvements to emerge. Our willingness for genericity, must be seen as a contribution in itself, in the sense of the facilitation for novel contributions.

Beyond this evident aspect, we could draw an interesting future for our framework. Although defining metrics is typically human compatible (and especially computer incompatible), simple parameters should not be set manually. As a spreading field, evolutionary robotics provides some exciting opportunities for the developmental robotics community, and could help us in this context. We mean that the very practical way of implementing evolutionary mechanisms consists in growing individuals with different genomes, and to make reproduce the best adapted ones in order to grow the next generation. This process can be reiterated as long as you may observe some interesting progress towards adaptation to the environment. That is why we truly believe that a possible *EvoFIMO* could constitute a very interesting perspective for the development of our research as well as for the whole community.

References

Baranes, A. and Oudeyer, P.-Y. (2010). Intrinsically-motivated goal exploration for active motor learning in robots: A case study. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*.

Baranes, A. and Oudeyer, P.-Y. (2013). Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73.

Barto, A. G., Singh, S., and Chentanez, N. (2004). Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of International Conference on Developmental Learning (ICDL)*. MIT Press, Cambridge, MA.

Blank, D., Kumar, D., Meeden, L., and Marshall, J. (2005). Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture. *Cybernetics and Systems*, 36.

Deci, E. L. and Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior (Perspectives in Social Psychology)*. Perspectives in social psychology. Plenum Press.

Hervouet, F. (2013). Autour de la preuve en intelligence artificielle. In *La preuve et ses moyens – Actes des journées interdisciplinaires de Rochebrune (sous presse)*.

Hervouet, F. and Bourreau, E. (2012). Improvement proposals to intrinsically motivational robotics. In *Proceedings of the second joint conference ICDL-Epirob'12*.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). Scipy: Open source scientific tools for python.

Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. In *In: ECML-06. Number 4212 in LNCS*, pages 282–293. Springer.

Meeden, L. and Blank, D. (2006). Editorial: Introduction to developmental robotics. *Connection Science*, 18(2):93–96.

Oudeyer, P.-Y. and Kaplan, F. (2004). Intelligent adaptive curiosity: a source of self-development. In Berthouze, L., Kozima, H., Prince, C. G., Sandini, G., Stojanov, G., Metta, G., and Balkenius, C., editors, *Proceedings of the 4th International Workshop on Epigenetic Robotics*, volume 117, pages 127–130. Lund University Cognitive Studies.

Schmidhuber, J. (1991a). Curious model-building control systems. In *Proceedings of the International Joint Conference on Neural Networks, Singapore*, volume 2, pages 1458–1463. IEEE press.

Schmidhuber, J. (1991b). A possibility for implementing curiosity and boredom in model-building neural controllers. In Meyer, J. A. and Wilson, S. W., editors, *Proceedings of the International Conference on Simulation of Adaptive Behavior*, pages 222–227. MIT Press/Bradford Books.

Scriven, M. (1991). *Evaluation Thesaurus*. SAGE Publications.

Steels, L. (2004). The autotelic principle. In *Embodied Artificial Intelligence*, pages 231–242. Springer.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Varela, F., Thompson, E., and Rosch, E. (1991). *The Embodied Mind: Cognitive science and human experience*. MIT Press, Cambridge.

Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., and Thelen, E. (2001). Autonomous mental development by robots and animals. *Science*, 291.