Berthouze, L., Prince, C. G., Littman, M., Kozima, H., and Balkenius, C. (2007).
Proceedings of the Seventh International Conference on Epigenetic Robotics: Modeling
Cognitive Development in Robotic Systems. *Lund University Cognitive Studies, 135*.

# Human-robot interaction based learning for task-independent dynamics prediction

**Jean-Charles Quinton**
Computer Science Research
Institute of Toulouse (IRIT)
2 rue Camichel BP 7122 - F
31071 Toulouse Cedex 7, France
quinton@n7.fr

**Tetsunari Inamura**
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo 101-8430, Japan
inamura@nii.ac.jp

## Abstract

Predictive dynamics learning can be drastically improved for robots by interacting with humans or other agents and taking advantage of their knowledge. This paper presents an initial work on interaction based learning algorithms embedded in a task-independent model adapted to continuous and multiple scale anticipations.

## 1. Introduction

Though anticipation of the environment dynamics is not commonly used in artificial intelligence, it is widely spread in the robotics field (Kawato, 1999). Any behavior requiring fast reactions and interactions with the environment can hardly cope with the computational delays introduced by the full processing of sensors information (O'Regan and Noë, 2001). Moreover when performing in real environment, noise hinders feature detection algorithms, but often, only vague actions progressively refined are enough to converge to a specific target or adapt to the situation.

Still in any complex environment like our own world, specially if it is changing or involves myriads of interactions between objects, particles, or even living beings, generating an exact global model seems like an utopia. Depending on the task or behavior, precise anticipations might be required at a local range, but this might only be learned by interacting and trying, directly linking perceptions and actions, and not specified by any a priori knowledge (Brooks, 1991). The problem in high dimensional spaces, taking in account perceptions and possible actions, is the uselessness and impossibility of anticipating everything anytime and anywhere.

To improve learning efficiency and not just wait for the correct action to be chosen then stored by trying randomly, interacting with other robots, agents or humans stands as a good solution. Instead of concentrating on mimesis (Inamura et al., 2004), this paper emphasizes the generalization/specialization capabilities of living organisms. By observing others actions or directly acting, the robot will continuously update its internal predictions, reevaluate its confidence in anticipations and communicate to refine its knowledge.

As stated and explained by the interactivist framework (Bickhard, 1993, Bickhard, 1996), objects and concepts can be further distinguished and derived from simple sensorimotor interactions. In this basic work, symbols and their semantics are introduced to simplify the robot/human communication but understanding the other should be integrated to the system evolution. From the robot point of view, both the human and the physical world are part of the environment but all of them can interact in different ways and at different abstraction levels. Interacting with the environment is like conversing with someone: acting is asking a question to the environment, anticipating the evolution of the dialog, based on assumptions about the other and the context. Perceiving is hearing and interpreting the answer, confirming or undermining the anticipations, revising our beliefs, learning and changing our behavior.

The work presented in this paper is an interdisciplinary attempt to adapt Piagetian schemes and interactivist networks to continuity and robotic constraints. Though results might be improved and the application extended in the future to more complex and realistic situations, this article details principles and algorithms embedded in the basic toolkit developed and tested on simple tasks.

The current approach is quite similar to Tani's neural networks models for navigation (Tani, 1996). Still when using reflexive internal activity, our view provides an homogeneous structure, thus resulting in an implicit hierarchy between both navigation and command interactions. We also have to refer to the recently published achievements of Oudeyer et al. that converge on the confidence and BSP algorithms.

These were introduced to represent the relative interest or curiosity in previously discovered but not mastered activities (Oudeyer et al., 2007). They are used to model exploration behaviors rather than for interpolation and optimization purposes. The authors also make a wide review of various machine learning approaches and goal-reaching/exploration theories.

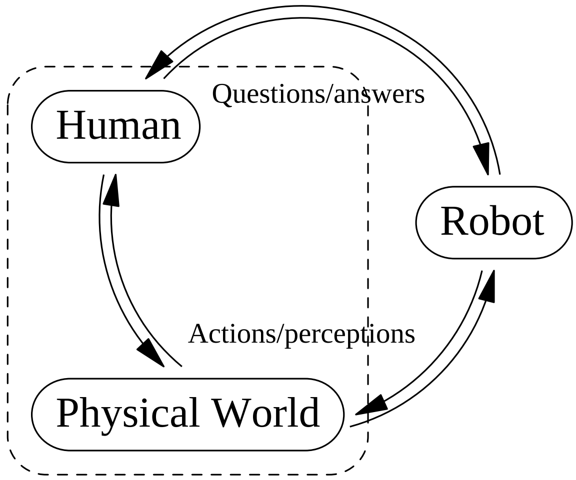


Figure 1: Interactions between the robot and its environment, composed by the human user and the physical world or simulator

## 2. Model

To apply the above principles and generalities to real-time control of robots, as well as to simplify the temporal anticipations, constant discrete steps are introduced between updates to avoid handling different time-scales. The update frequency might correspond or be lower than the sampling rate from sensors and command rate to actuators. A function $f$ of $\mathbb{R}^{p+a} \rightarrow \mathbb{R}$ is defined for each anticipated variable, where $p$ and $a$ are respectively the number of perception and action dimensions. As for the simple application detailed later in the performance section, arbitrary dimensions can be abstrations (such as the position or speed of objects), but might be direct input from any sensor.

For every time step, this function anticipates the next value of the chosen variable from the current state of sensors and taken actions. In some sense, it models the regularities and laws of the environment (Gibson, 1979). Depending on the task to perform, the set of anticipated variables might vary, the influence of the left elbow rotation might be neglected when dealing with grasping in the right hand for instance. The robot will be considered adapted if it succeeds performing tasks by approximating the real function, which might be of infinite complexity when taking the smallest interaction into account. No full understanding of the world physics, properties and people intentions is mandatory to navigate correctly in a crowd.

This model supposes the world is deterministic or the robot has access to all necessary information to anticipate correctly, the same actions taken from the same state at different times will lead to the same immediate evolution of the system. This does not mean the robot needs data over the full environment, but that it can perceive accurate information for its local behavior.

One might notice the use of real values for perception and actions; any complex input or output must then be decomposed into real values. The robot has no a priori knowledge about the required detail or range for each dimension, therefore $\mathbb{R}$ is used for each variable. Though an angle will be clamped into $[-\pi; \pi]$ or an applied force limited by physical constraints, the robot can perfectly learn within the reachable limits and sometimes extrapolate for outer values.

Robots have in general access to a wide range of sensors, and it is useful to reduce the set of variables by selecting relevant ones. Human beings in everyday life do not explore the gigantic space of reachable situations, but coordinate well-known interactions into more complex ones. At any point during ontogenesis, each behavior tries to assimilate the current situation (Piaget and Inhelder, 1969) then requires higher precision and exact anticipations when errors lead to a greater divergence from target states. Behavior is defined by local regulated patterns of activity within a huge network of interactions. For example writing with the right hand requires great dexterity in the control of the fingers to move the pen on the paper, but the left hand only requires rough commands to press the sheet.

During the development or learning of an activity, the confidence in actions taken increases with the practice and confirmation of the anticipations. In the following sections, confidence is introduced to reflect the level of expertise the robot has in local areas of the state space. If an area has been totally an perfectly explored, i.e. with low errors relatively to current task requirements, it should be preferred to perform without taking risks, but will not require further training. A typical example is the child learning to throw a ball (Piaget, 1952).

## 3. Implementation

### 3.1 Anticipation surface

For each anticipated variable, the associated approximated surface is defined by references points. Points correspond to past experiences memorized by the robot. They are composed of the perceptions and

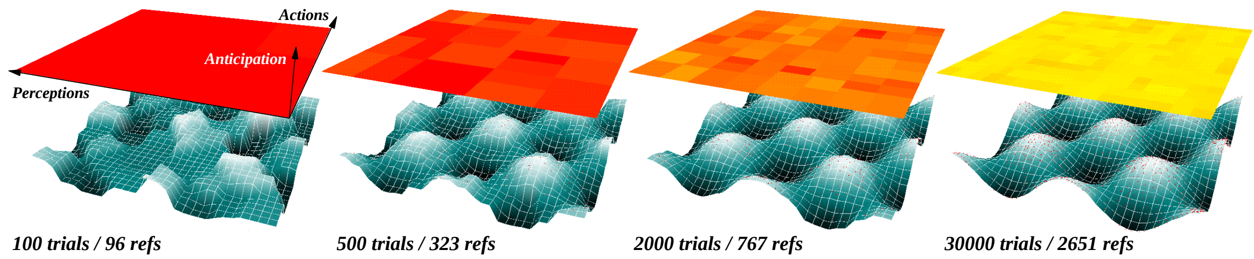


Figure 2: Interpolated surface for an increasing number of states experienced: the higher the number of trials, the higher the confidence and the lower the percentage of points used as references. The confidence is displayed as a red to yellow gradient for each area in the BSP tree. Though the dimensions have here no meaning, the function approximated being a simple cos(x)cos(y), each corresponds in the learning algorithm to a perception or action, and the image axis to the anticipation.

actions at time $t_n$ and the change in the value of the anticipated variable at $t_{n+1}$. Storing only differences allows having smaller values, and initialization to 0 corresponds to the assumption of constancy by default, i.e. the surface is a null plane until changes are observed.

Since the robot might need to anticipate at any point in the state space, the dynamic environment always being in a slightly different configuration, interpolation is performed between reference points. This allows generalization from observed elements. Inverse Distance Weighting (IDW) is used and might be improved by using Shepard's method (Shepard, 1968). In the following formulas $f$ is the exact function observed by trials and $f_a$ is its approximation. Considering $n$ reference points $(r_i, a_i) \in \mathbb{R}^{a+p} * \mathbb{R}, \forall i \in [1; n]$ where $a_i = f(r_i)$, the simplest expression of IDW interpolation at state $s$ is:

$$f_a(s) = \frac{\sum_{i=1}^{n} w(r_i, s) * f(r_i)}{\sum_{i=1}^{n} w(r_i, s)} \qquad (1)$$

A simple pondered sum where $w_i$ is the weight associated with the $i^{th}$ reference point computed as follow:

$$w(r_i, s) = \frac{1}{\|r_i - s\|_2} \qquad (2)$$

Though the $l^2 - norm$ is written here, various expressions for the weighting have been experimented but will not be thoroughly detailed in this paper.

These equations asserts the resulting surface will be continuous and will pass through all reference points. Indeed when interpolating at a reference state $r_i$, the associated weight reaches $+\infty$ and $f_a(r_i)$ is equivalent to $\frac{w(r_i,s)*f(r_i)}{w(r_i,s)}$. Numerical adjustments in the source code account for this property without handling infinite values.

Points can be arbitrarily spread in the state space and a regular discrete grid would not reflect the various needs in density and precision, conducing to either a huge reference point database (which is an issue for access and memory) or a rough approximation (the robot being unable to perform tasks requiring high dexterity at some point).

### *3.2 Reference states*

Depending on the tasks to perform, a highly heterogenous repartition of references might be observed and if tasks cover very different skills, the number of points might drastically increase. Therefore structures and algorithms for fast search, retrieval and insertion have been embedded to model the state space though introducing limitations in the good properties of the original interpolated surface.

For logarithmic complexity access and insertion of points, a Binary Space Partitioning tree is implemented. For a review and original use of BSP Trees for 3D representations, see (Fuchs et al., 1980). Its root corresponds to the full space and its nodes to a split in the state space. Leaves are associated to sets of points whose size is kept under a given limit by the tree algorithms. These have been developed and optimized for fast leaf splitting, properties updates, incremental search and memory management, as to generate and modify the overall tree representing the whole robot experience in real-time.

One of the main interests of this BSP structure is the selection of a reduced set of points for interpolation. A search algorithm returns all the leaves intersecting a "sphere" defined by its center, the point at which the interpolation will take place, and a radius for each dimension ($\in (\mathbb{R}^{a+p}, \mathbb{R}^{a+p})$). Properties associated with the nodes and leaves include a bounding box around the points, to limit the number of leaves returned without any points in the specified ball.

The IDW interpolation process is modified as follows. A first incremental search returns the closest reference to the state where interpolation is performed. Based on the minimal distance to references, new radiuses are then computed to take into account all points that cannot be neglected compared to the

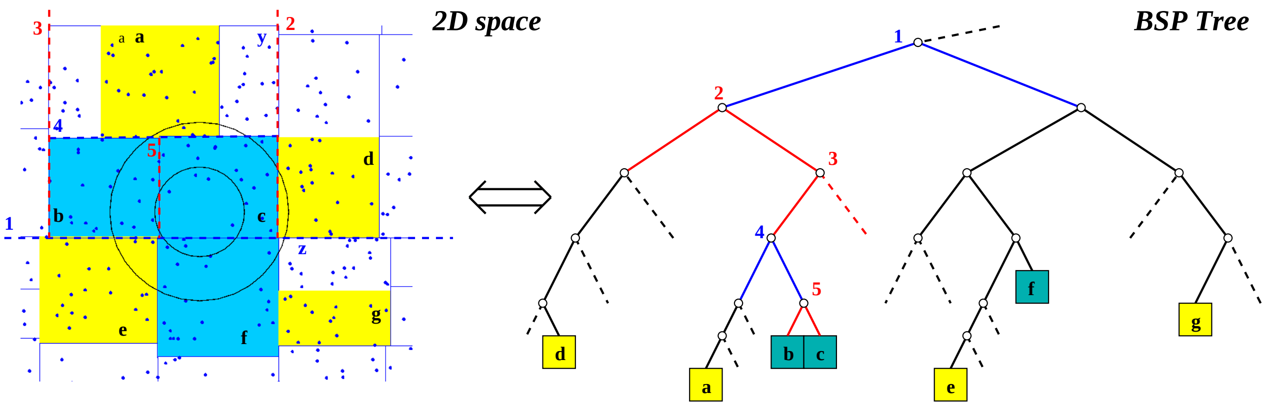


Figure 3: The dashed lines represent splits in the partial space represented. Only a portion of the nodes (splits) and leaves (areas containing points) are displayed. The two circles correspond to the incremental search parameters to guaranty a correct interpolation. The returned leaves are filled in blue and yellow. The reader might notice the lack of reference points near the center which required the increase in the search radius. Though the largest circle intersects the areas labeled y and z, they are not selected due to algorithmic optimizations since no point lies inside the circles.

closest one. This introduces minor discontinuities when interpolating close to leaves borders and can not account for points shadowing each others, but in general reduces the computational cost dramatically.

### *3.3 Confidence in anticipations*

The other main interest lies in the possibility to associate confidence in anticipations with the areas defined by the tree nodes and leaves. It is much harder with points since the environment dynamics almost never goes through the exact same state and interpolating confidence might be quite cumbersome. Therefore confidence is computed based on a ratio between errors committed in the associated area and correct anticipations:

$$confidence = max\left(0, 1 - \frac{nb_{errors} + k}{nb_{trials} + 1}\right) \qquad (3)$$

Confidence is clamped in $[0;1]$ and $k$ is a constant defining the number of trials to occur before it can reach strictly positive values. This is to avoid getting a high confidence in newly discovered areas with a few lucky trials.

## 4. Algorithms

### *4.1 Learning*

Learning runs at all times by comparing the anticipated state at $t_n$ (from $t_{n-1}$ perceptions and taken actions) and the actual variables values perceived at $t_n$. If the error committed is above a certain relative threshold, the local interaction performed ($t_{n-1}$ state + $t_n$ variables values) is added as a new reference point for the interpolating surface.

If the error triggers an insertion, the confidence in the local associated area in the BSP tree is decreased by incrementing $nb_{errors}$, and increased if the difference is considered acceptable since $nb_{trials}$ is incremented anyway.

### *4.2 Goal reaching*

The goal reaching algorithm implemented tries to go straight to a given goal from current state at any time. It selects the best actions to take so that the agent gets closer to the goal avoiding large deviations and very low confidence areas where behavior is unknown. In practice the collinearity between the anticipated changes vector and the current state-goal vector is maximized. Excluding the confidence and distance pondering, the action can be explicitly selected as :

$$action(s, g) =$$

$$argmax_{act \in \mathbb{R}^a}\left(\frac{anticip(s_{act}).(g - s)}{\|anticip(s_{act})\|_2 \|g - s\|_2}\right) \qquad (4)$$

$$anticip(s_{act}) = \left[f_{a_{perc}}(s_{act})\right]_{perc \in \mathbb{R}^p} \qquad (5)$$

In these expressions, $s$ is the current state, $s_{act}$ the current state where actions have been set to *act*, and $g$ the goal to reach. The number of dimensions for atomic actions and anticipated perceptions respectively are $a$ and $p$. $anticip(s_{act})$ is simply the anticipated perceptions vector in $\mathbb{R}^p$. In the current implementation, goal can be partially determined by not fixing all the perceptions for the target state.

Though interactions might be quite fuzzy, taking into account uncertainty and noise on actions and perceptions, the algorithm will cope with errors and converge to the goal. Still if the task requires stepping through far from straight line states, it will probably fail reaching the goal due to time constrains and spontaneous dynamics of the environment attracting the system to the closest local equilibrium or low energy area.

The algorithm presented here is a very simple one, that might and should be regulated by higher level or wider range processes. These could modulate the activity of local interactions and the confidence in anticipations to change the dynamical landscape and specially its attractors, modifying the trajectory within the state space. This would also introduce notions such as habituation, curiosity or motivation at multiple scales. Scales and the resulting hierarchy might be implicit but defined by the internal activity of concurrent processes, more abstract but derived from the sensory-motor level as often described in neuromodulation theories (Dehaene et al., 2006).

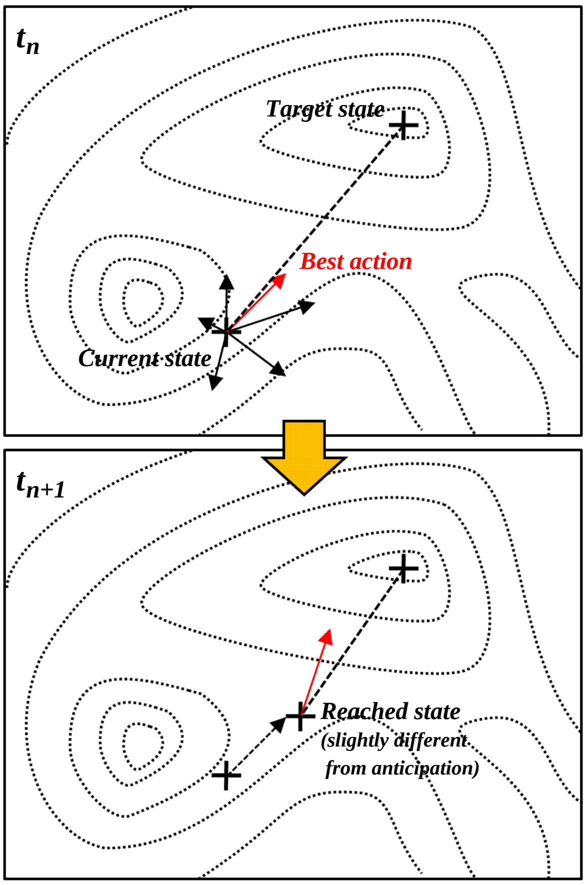


Figure 4: For each time step, the best action (confidence + collinearity criterion) is selected and applied by the robot. Anticipated change vectors are displayed as arrows from the current state. The dotted lines represent level sets of the anticipated variable surface. The resulting state might differ from anticipation but the process will go on until it converges to the target.

### *4.3 Exploration*

To improve the knowledge the robot has of its environment dynamics, it might explore areas where it has low confidence in predictions. This autonomous exploration is linked with the notions of curiosity and motivation. To avoid the problems like being drawn to completely new areas, maybe not even possible to reach or without any interest in the current tasks context (Schmidhuber, 2006, Stout et al., 2005), the robot performs in low confidence areas that lies within its effective trajectories.

Even if the learning of the dynamics is task-independent, the robot generalizing and interpolating interactions, the local levels of details and history of the system are goal-directed. As long as precision in anticipations is not needed or the sub-area considered is not even crossed during the underlying task performance, no refinement is necessary. For example one will not learn to control a pen with both hands except he tries to and will master writing only with the dominant hand.

## 5. Interactions

Random exploration might take a very high number of iterations to reach a goal, explicitly given or implicitly set. Even by constantly improving anticipations while making mistakes, the probability to reach the target decreases with the number of dimensions.

### *5.1 Human/robot interactions*

To fasten the learning process of an adapted behavior relative to a specific task, human/robot interactions help a lot. Communication between human and robot can be of quite a high level, using sentences or speech recognition, but direct guidance by acting on the environment can be the simplest way of interacting with an agent certainly having different cognition mechanisms.

The human user is in general supposed to be competent or at least able to perform the task better than the robot at its current learning state. In a way, he teaches the robot how to reach a higher skill level in a cognitive apprenticeship way (Vygotsky, 1930). Anyway if it fails at the task or is only able to successfully perform half of it, this will not hinder the learning process. The robot will just improve along with the user trying to help. As this kind of social interaction process goes on, it can be regulated by questions and answers on both sides, for example to confirm actions. Even if both the robot and the human teacher can act anytime on the environment, therefore potentially cooperating on any task, the human has priority in case of conflicting decisions.

### 5.2 Human control

The human interacts with a simulated environment and the robot through a GUI and the mouse. The interface displays the physical system in 2D where the user directly clicks on the cart target position. Though different interactions have been explored, a second order regulation on requested position and cart speed seemed the more intuitive to the human experimenters. The force applied is computed as:

$$force = ((mx_t - x_t) - v_t/k_1) .k_2 \qquad (6)$$

$m_x$ stands for the mouse position and $x_t$ the current cart position in screen coordinates. $k_1$ and $k_2$ are parameters to tune the sensitivity and behavior. This interaction is a lot different from the direct force computation performed by the robot. As human beings can exchange and communicate without sharing the same history or exact assumptions on the situation, the robot only learns by observing and not inferring on the potential user knowledge or skill. To not confuse the user and environment forces, the robot feels the teacher's commands as if it was actually performing the action by itself. This behavior is similar to an adult guiding a child by holding and moving his hand to correct a movement or writing. The only difference is whether the robot has the choice of action.

## 6. Performances

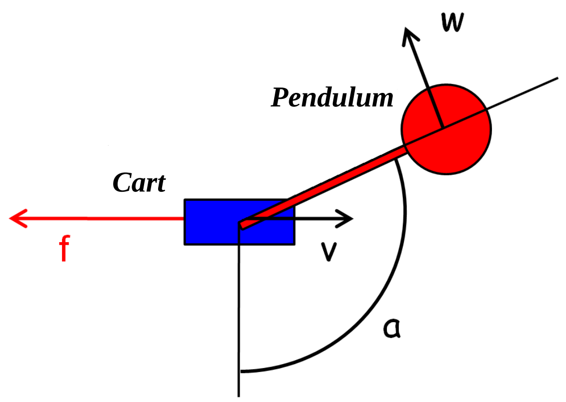


Figure 5: A force $f$ can be applied on the cart and perceived in a proprioceptive way. The robot also gets the cart speed $v$, pendulum angular speed $w$ and angle $a$.

To evaluate the algorithms, a simple cart-pendulum simulator has been developed using Java (JDK 1.6) on a notebook computer equipped with an Intel Pentium M735. The interactions (through a window displaying the physical system and a dialog frame for the human) being processed in real time, the performances of the learning algorithm do not really depend on the computational power. Nevertheless it is relevant to evaluate the efficiency of the BSP tree and interpolation algorithms. These can be easily parallelized and executed on modern graphics card using shaders if the final robot control application requires additional free processing power.

Though the physics takes into account friction, rod tension, track reaction, gravity forces and more, the robot only perceives the cart speed, pendulum angular speed and position. A single force is applied to the system to change its spontaneous dynamics and keep it far from equilibrium. It is limited to prevent the robot from using very strong impulsions to artificially reach the goal faster with an impossible behavior in any real environment. The physics engine uses a Runge-Kutta approximation to integrate the differential equations with a 10ms time step.

### 6.1 Algorithms computational complexity

The figure 8 displays the logarithmic and linear complexities of the algorithms to add new references to the BSP tree and interpolate in the state space. The irregularities come from memory reallocations, object creations and splits in the tree. These have not been filtered or averaged to show that the computation time of the whole process of interpolating anticipations and updating the trees remains under the millisecond. Moreover the linear complexity for the interpolation tends to become logarithmic for higher numbers of points (above 30000), due to the algorithmic optimization in the number of references used.

### 6.2 Real-time goal reaching performance

The requested task for performance evaluation is a stabilization of the pendulum in the lower position. This allows comparison between the spontaneous dynamics of the system without any force applied, a random application of forces within the allowed range, the human and the robot performances depending on previous training/teaching.

The following table shows the results from different initial states of the pendulum and various control types. The initial cart speed is set to 0 for all the conditions. Reaching the perfect equilibrium being impossible even for a human being, the task is fulfilled when the current state enters a sphere around the goal state ($|a| < 0.1\ rad$, $|w| < 0.2\ rad.s^{-1}$, $|v| < 0.1\ m.s^{-1}$).

When applying a random force in the allowed range, the system never reaches a state close to equilibrium. This is just to confirm that the application of the right force amount with the right timing is necessary to fasten the stabilization process. For example the robot rapidly learns to apply a strong opposite force when the pendulum speed is maximal, or to alternate left and right pushes on the cart to limit $v$ variations and their amplitudes.

The robot exploring the dynamics by itself starting from scratch takes a lot of time before learning a satisfactory behavior. By trying random actions and progressively improving the anticipations, it is less efficient than the spontaneous convergence to equilibrium when no force is applied on the system. A detailed comparison between the human learning to manipulate the cart through the interface, and the robot getting trained and interacting with the human to improve its skill can be found on figure 7.

Some of the initial situations might be similar for the spontaneous behavior of the dynamical system, but can lead to totally different evolution under the robot control depending on the previous experience. For example with the symmetric $a = -\frac{\pi}{4}$, $w = 0$ and $a = \frac{\pi}{4}$, $w = 0$ initial states, the robot is very efficient only on the situation where it practiced a lot, otherwise the anticipations are inaccurate and the convergence occurs with an additional learning. This is partly due to a lack of notions such as similarity in the assimilation process: only references with close variable values are used.

| Initial state \ Force | a | $\frac{\pi}{2}$ | $\frac{\pi}{2}$ | $\frac{\pi}{4}$ | $-\frac{\pi}{4}$ | $-\frac{\pi}{4}$ |
|---|---|---|---|---|---|---|
| | w | 0 | 4 | 0 | 0 | 2 |
| none (spontaneous) | | 26 | 30 | 18 | 18 | 22 |
| applied randomly | | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ |
| expert human | | 4 | 6 | 3 | 3 | 2 |
| new robot | | 81 | 96 | 27 | 36 | 47 |
| 5 minutes old robot | | 8 | 12 | 3 | 8 | 13 |

Figure 6: Mean time to satisfy the task ending condition (in seconds) to compare real-time performances with different types of interactions. The initial angle and speed respectively are in rad and rad/s.

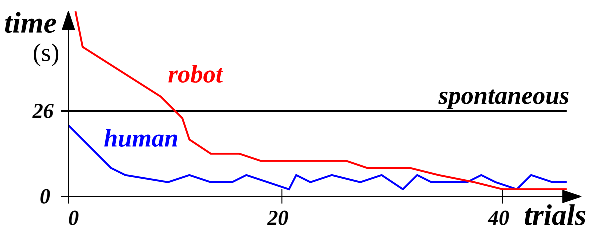


Figure 7: Comparison between robots and human users new to the controls. The spontaneous evolution to equilibrium is given as a reference. ($a = \frac{\pi}{2}$, $w = 0$, $v = 0$).

## 7. Discussion and perspectives

Even if the algorithms are optimized to handle a huge set of reference points and interpolate on a high dimensional space, the combinatory explosion during exploration prevents from using such basic method on realistic complex behaviors. For instance the robot used for tests was a Hoap3 humanoid robot from Fujitsu, reading position and sending commands to its 28 joints, getting information from 10 touch sensors, 2 video cameras and more.

More interactions not covered by this paper have been introduced to improve and guide the robot exploration of the state space. Simple dialog between the robot and the human based on human understanding of the situation and robot analysis of its relative confidence on different scales will help limiting the dimensions to the locally relevant ones (as discussed in section 2). Human advice might have a direct effect on the robot algorithms or simply be integrated to the framework as other anticipated variables. In this case the robot would learn human goals and intentions in a naturalistic communication process, correlating its actions with the environment evolution and human response.

For more complex tasks involving non obvious intermediate steps, anticipations at different timescales would be helpful. Long-range slow processes would roughly give general direction and goal (for example going somewhere) while fast local regulations would anticipate for little variations in the environment (avoiding obstacles and keeping balance). Introduction of time as a standard anticipated variable would allow the robot to represent invariant principles as well as to keep trace of specific events. This would also allow the learning to occur in a non fully observable environment from the robot point of view. The uniqueness of the image on the anticipated surface would not be a problem anymore, the time flowing constantly.

Finally one can easily take advantage of this model to evaluate necessary sampling rate for sensors and actuators relative to a given task. If the local complexity of the surface is too high, i.e. errors are still committed even when reducing the delay between samples to improve precision in anticipations, the task constraints should be relaxed or sensors changed. Nevertheless this kind of learning depending on the properties of the perceptual apparatus as for humans and their physical body, it would have to be started over in case of fundamental change.

## 8. Conclusion

This paper presents a basic toolkit that must be completed by new algorithms and concepts to be applied to robots navigating in real environments. Still the BSP tree is easily customizable and various properties for exploration or learning can be attached to it. This is a work in progress that is extended to handle multi-scale anticipations on any dimension to provide improved methods for goal-reaching and regulated interactions.

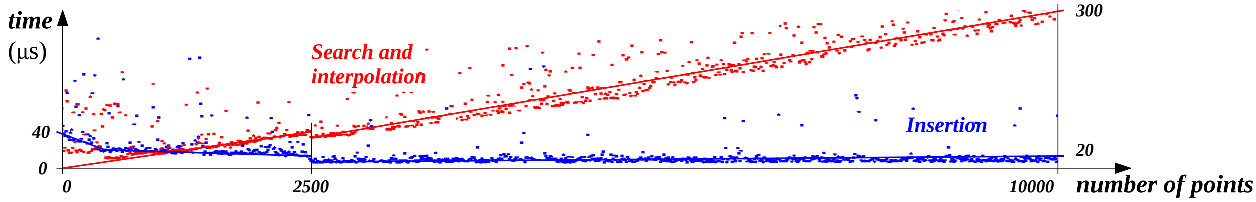


Figure 8: The computation performance for the access and insertion of a single point in the BSP tree is shown in blue. The interpolation process (finding the leaves and iterating on the points properties) is displayed in red. Both are expressed in $\mu s$ and depend on the number of points present in the tree structure. Of course the points repartition in the state space has a great influence but the tree rapidly becomes locally balanced.

The structures and algorithms might be greatly modified to take into account number of concepts and properties of the interactivist framework to develop more interesting and realistic applications. Even in their current version, they display simple human-like learning capabilities and adaptability to various precision requirements for a relatively low computational cost. This is mainly due to the differential predictive model associated with human-robot-environment interactions. Advices, acknowledgments, corrections or symbol association and recognition will be included to increase the shared knowledge between agents and take the most of the behavioral interactions.

## 9. Acknowledgments

The three months internship this work is part of was made possible by the Memorandum Of Understanding (MOU) agreement between the Japanese National Institute of Informatics (NII) which supported the internship and the National Polytechnic Institute of Toulouse (INPT).

## References

Bickhard, M. H. (1993). Representational content in humans and machines. *Journal of Experimental and Theoretical Artificial Intelligence.*

Bickhard, M. H. (1996). The emergence of representation in autonomous embodied agents. *AAAI Fall Symposium on Embodied Cognition and Action.*

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence.*

Dehaene, S., Changeux, J.-P., Naccache, L., Sackur, J., and Sergent, C. (2006). Conscious, preconscious, and subliminal processing: a testable taxonomy. *Trends in Cognitive Sciences.*

Fuchs, H., Kedem, Z. M., and Naylor, B. F. (1980). On visible surface generation by a priori tree structures. *ACM Computer Graphics.*

Gibson, J. J., (Ed.) (1979). *The ecological approach to visual perception.* Houghton Mifflins, Boston.

Inamura, T., Nakamura, Y., and Toshima, I. (2004). Embodied symbol emergence based on mimesis theory. *International Journal of Robotics Research.*

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology.*

O'Regan, J. and Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences.*

Oudeyer, P.-Y., Kaplan, F., and Hafner, V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation.*

Piaget, J., (Ed.) (1952). *The Origins of Intelligence in Children.* International Universities Press.

Piaget, J. and Inhelder, B., (Eds.) (1969). *The Psychology of the Child.* Basic Books, New York.

Schmidhuber, J. (2006). Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science.*

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the ACM National Conference.*

Stout, A., Konidaris, G. D., and Barto, A. G. (2005). Intrinsically motivated reinforcement learning: A promising framework for developmental robot learning. *Proceedings of the AAAI Spring Symposium on Developmental Robotics.*

Tani, J. (1996). Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Trans. on Systems, Man, and Cybernetics Part B: Cybernetics.*

Vygotsky, L. S., (Ed.) (1930). *Mind and Society.* Harvard University Press.