

# Modélisation du comportement animal et apprentissage par renforcement

Philippe Preux  
Samuel Delepoulle  
Laboratoire d'Informatique du Littoral (LIL)  
UPRES-JE 2335  
Université du Littoral Côte d'Opale  
B.P. 719  
62228 Calais Cedex, France  
`{preux,delepoulle}@lil.univ-littoral.fr`

Jean-Claude Darcheville  
Unité de Recherche sur l'Évolution des Comportements  
et des Apprentissages (URECA)  
UPRES-EA 1059  
Université de Lille 3  
B.P. 149  
59653 Villeneuve d'Ascq Cedex, France  
`darcheville@univ-lille3.fr`

Rapport LIL-01-02

Novembre 2001

## **Abstract**

L'apprentissage par renforcement fournit un cadre explicatif de nombreux aspects du comportement animal, en particulier de processus sophistiqués comme l'acquisition d'un nouveau comportement, l'adaptation du comportement à l'environnement, l'acquisition de séquence comportementale, la généralisation de l'apprentissage, l'extinction, le façonnage, les interactions sociales, l'éducation des petits, ... On peut donc espérer que sa formalisation algorithmique soit utile pour résoudre certains problèmes d'informatique ou de robotique. Ce type d'apprentissage fournit des idées qui méritent d'être étudiées par les informaticiens pour qui apprentissage par renforcement rime trop souvent avec une procédure d'apprentissage par essai/erreurs. Dans cet article, nous présentons et discutons de la

formalisation algorithmique de l'apprentissage par renforcement. Nous présentons ensuite une architecture multi-agents dont le contrôle de chacun des agents est réalisé par un algorithme de renforcement. Ce travail a pour objectif d'évaluer les possibilités d'une approche purement renforcement, d'une part comme modèle du vivant, d'autre part pour évaluer les possibilités de ce type d'approche en terme d'algorithme de résolution de problème d'informatique. Enfin, nous discutons le développement de ces techniques pour les systèmes artificiels.

Reinforcement learning provides a framework to explain numerous aspects of animal behavior. Among others, sophisticated processes are concerned such as the acquisition of a new behavior, the adaptation of behavior to its environment, the acquisition of behavior sequences, generalisation, extinction, shaping, social interactions, ... We can hope that an algorithmic formalization may help solve problems in computer science, and robotic. This kind of learning provides ideas that deserve to be studied by computer scientist who consider too often that reinforcement learning equals a mere trail and error learning procedure. In this paper, we present and discuss how reinforcement learning may be brought into an algorithmic form. Then, we present a multi-agent architecture in which each agent behavior is controlled by a reinforcement algorithm. This work aims at assessing a purely reinforcement-based approach as a model for the behavior of living beings, and to assess its possibilities as a resolution algorithm in computer science. Finally, we discuss the development of these techniques for artificial systems.

**Mots-clés** : apprentissage, modélisation, dynamique du comportement

# 1 Introduction

Après avoir envisagé de produire des systèmes artificiels passant le test de Turing avant la fin du siècle dernier, certains d'entre-nous visent aujourd'hui des objectifs plus modestes (du moins en apparence) : reproduire les capacités des animaux (insectes, ...) à survivre et vivre dans leur environnement. Aussi, une communauté de chercheurs en informatique cherchent-ils à s'inspirer de la structure et du comportement des animaux, autrement dit, à modéliser le vivant. Concernant le comportement, il peut sembler raisonnable de commencer par étudier le travail des spécialistes du comportement animal<sup>1</sup>. Ainsi, la psychologie du comportement s'est intéressée aux facteurs responsables de l'évolution du comportement animal ainsi qu'à l'effet de cette évolution. Contrairement à un malentendu courant, l'ambition de ce projet de recherche n'est pas de se cantonner à un modèle simpliste de type stimulus-réponse (S-R). Le projet est d'envisager l'ensemble des conduites adaptatives d'un organisme (ou d'un groupe d'organismes) dans un environnement réel, donc dynamique.

L'idée de processus de renforcement a été proposée pour rendre compte des observations sur l'évolution des comportements et des apprentissages qui avaient été menées depuis longtemps. Par exemple, à la fin du XIX<sup>e</sup> siècle, Thorndike a proposé la loi de l'effet  $z$  : la probabilité d'apparition d'un comportement augmente s'il est suivi de conséquences favorables pour l'organisme [74, 75].

Cela dit, l'objectif n'est pas de s'en tenir au processus de renforcement lui-même mais plutôt de comprendre l'ensemble des relations complexes entre les comportements, l'environnement dans lequel ils s'observent et leurs conséquences à plus ou moins long terme. La contingence de renforcement  $z$  [65] est alors une entité qui désigne les relations entre la réponse et les événements de l'environnement, ceux qui précèdent et qui ceux suivent cette réponse.

À la suite de Thorndike, Skinner a donc proposé l'idée selon laquelle les comportements d'un animal sont sélectionnés (= renforcés) en fonction des conséquences qui ont suivi leur émission dans le passé de ce même animal [66]. De nombreux processus complexes ont été étudiés dans ce cadre aujourd'hui très actif [68] : processus de discrimination, de généralisation, des chaînes complexes de comportements, du renforcement différé, de comportements sociaux [36] ou symboliques, de l'adaptation temporelle [46], des mécanismes de variation du comportement [16] et des comportements verbaux [13]. En résumé, le terme d'apprentissage par renforcement désigne dans cette communauté une grande variété de phénomènes généralement complexes.

Nous avons donc là des modèles rendant compte de la dynamique comportementale animale, notamment de processus complexes dont nous aimerions bien que nos artefacts soient dotés. Dans ce cadre, des travaux sont menés en informatique et robotique, en collaboration avec des chercheurs en comportement animal. Dans cet article, nous entendons rendre compte brièvement de ces travaux en mettant l'accent sur la modélisation elle-même du comportement.

---

<sup>1</sup>dans l'ensemble de cet article, nous considérons l'Homme comme faisant partie du règne animal

Dans la suite de cet article, nous présentons tout d'abord les motivations pour simuler le comportement animal et les différents types de modèles algorithmiques utilisés à ce jour. Dans la section 3, nous mettons l'accent sur la classe de modèles algorithmiques d'apprentissage par renforcement dits à base de différence temporelle. Ensuite, la section 4 décrit et discute un modèle récemment proposé d'architecture multi-agents non supervisée où le contrôle des agents est exclusivement fondé ce type d'algorithmes. Nous indiquons en quoi ces modèles diffèrent des modèles précédemment discutés, les motivations et le résultat des simulations réalisées à ce jour. Enfin, la dernière section propose une discussion générale de ces travaux et leurs perspectives.

## 2 La modélisation du vivant

Dans cette section, nous discutons les motivations de la modélisation du comportement animal. Nous continuons par un rapide tour d'horizon de l'existant dans le domaine de la modélisation du vivant à base de modèles algorithmiques.

### 2.1 Pourquoi modéliser le vivant ?

Avant de nous pencher sur les techniques de modélisation du comportement du vivant, nous en rappelons brièvement les motivations. Ainsi, deux points de vue se dégagent :

- l'éthologue, le psychologue, le psychophysiologiste... y voient un moyen d'améliorer leur compréhension du vivant. La formalisation par un modèle algorithmique du comportement animal est intéressante en soi ; l'implantation sous forme algorithmique oblige à expliciter très précisément ce modèle ; la simulation permet alors de valider le modèle. Deux démarches se complètent alors : l'une consiste à analyser un système vivant pour en construire un modèle, tandis que l'autre consiste à animer et mettre à l'épreuve ce modèle pour le critiquer et, éventuellement, le valider. Dans cette deuxième démarche de synthèse du vivant, les simulations informatiques constituent un outil irremplaçable ;
- l'informaticien y voit un exemple et une source d'information de ce qu'il souhaite réaliser de manière artificielle : une entité autonome dont le comportement est adapté et s'adapte à son environnement, *a priori* inconnu (démarche *animat* [50]). Le souhait de réaliser une intelligence artificielle passe alors par la reproduction de la vie, la vie artificielle, du moins, de certaines caractéristiques clés de la vie (voir par ex. [77, 67, 10, 58]). Alors que le point de vue de l'intelligence artificielle classique a été de reproduire un cerveau, le point de vue de la vie artificielle est de reproduire un organisme vivant, les capacités cognitives en étant un résultat corollaire.

## 2.2 Comment modéliser le vivant ?

Nous nous penchons maintenant sur différentes techniques de modélisation du vivant ayant pour objectif de mieux le comprendre. Nous insistons sur le fait que nous ne nous intéressons pas ici à la modélisation du vivant pour résoudre des problèmes. Ce second point de vue est souvent adopté en informatique et dispose d'une littérature conséquente. Par ailleurs, nous mettons l'accent sur les modèles informatiques (à base de traitements symboliques) plutôt que mathématiques (à base d'équations différentielles). Nous ne faisons que survoler ce champ de recherche : une revue exhaustive de ces travaux nécessiterait en effet un très long article.

De nombreuses techniques et outils informatiques sont disponibles et ont été utilisés depuis 50 ans. Les motivations ont été diverses : à un niveau très cognitif, il peut s'agir de modéliser des processus décisionnels dans des organisations ; il peut aussi s'agir de modéliser des structures émergentes de comportements d'animaux sociaux, voire, l'activité de composés biochimiques (virus, bactéries, ...) en interaction avec les cellules d'un organisme. Ces deux cas forment les extrêmes du spectre des modélisations du vivant.

La modélisation de systèmes de prise de décision et d'organisation humaines (entreprises, sociétés, ...) utilise majoritairement des systèmes multi-agents basés sur des agents cognitifs qui disposent d'une connaissance, de représentations de leur environnement, de croyances, de règles de comportement, ... plus ou moins figées [25]. Ce type de modèles peut être utilisé pour appuyer une étude pluridisciplinaire d'un système (voir [11, 8] pour un exemple de ce type d'approche). Ces techniques sont notamment utilisées pour les études d'aménagement du territoire.

À l'autre bout du spectre, les systèmes multi-agents (SMA) à base d'agents réactifs sont très utilisés. Faisons remarquer que le qualificatif réactif qui peut faire penser qu'il s'agit d'agents qui agissent de manière purement réflexe peut tromper le lecteur. En fait, la véritable différence entre un agent cognitif et un agent réactif est finalement la manière dont la connaissance est représentée : elle l'est sous forme explicite de symboles (objets, frames, scripts, ...) représentant des règles, des croyances, des intentions, ... dans les agents cognitifs ; si certains agents réactifs réagissent aux stimuli de manière purement réflexe par un mécanisme codé et immuable dans l'agent, d'autres adaptent leur comportement en fonction des stimuli reçus et de leur histoire. Ces agents ne sont pas cognitifs car leur connaissance n'est pas représentée au sens indiqué plus haut, mais ce ne sont pas de simples architectures réflexes ; ces agents sont mêmes, pour certains, bien mieux à même d'adapter leur comportement et d'apprendre au cours de leur existence que des agents cognitifs dont le comportement est régi par des règles figées, qui en font finalement, de véritables agents stimulus-réponse. Donc, le point important n'est pas dans la manière dont est représentée la connaissance (explicitement ou implicitement), mais dans les capacités d'apprentissage et d'adaptation de l'agent. Dans la modélisation du comportement animal, c'est bien cela qui nous importe.

Pour commencer avec des agents purement stimulus-réponse et n'ayant pas

de capacité d'adaptation, on pourra citer les réseaux d'automates cellulaires. Par exemple, ceux-ci ont été utilisés pour modéliser le développement de la population de l'algue *caulerpa taxifolia* en Méditerranée [37].

Plus complexes que les réseaux d'automates cellulaires, les réseaux de neurones ont été étudiés pour explorer le fonctionnement du cerveau. Les travaux sur les réseaux de neurones formels ont en tout cas proposé des évidences expérimentales montrant qu'un réseau de neurones peut apprendre la réponse à fournir pour des entrées données, il peut montrer des capacités de généralisation de l'apprentissage, des cartes topologiques peuvent se constituer, ... Si les réseaux de neurones supervisés sont peu réalistes, d'autres réseaux proposent un modèle précis et réaliste du fonctionnement de nos neurones (voir par ex. les travaux de G. Edelman et son équipe, en particulier [24, 23]).

Les algorithmes génétiques pour lesquels l'objectif initial était de comprendre et modéliser la sélection naturelle [38, 39] ont donné lieu par la suite aux systèmes de classeurs [40], aux systèmes écologiques [41, 30] et à la simulation du système immunitaire [4, 29]. Les systèmes écologiques ont montré que des interactions complexes peuvent émerger entre des entités évoluant selon un schéma inspiré de la sélection naturelle, interactions telles la course aux armes biologiques où deux espèces complexifient continuellement leurs moyens d'attaques et de défense l'une par rapport à l'autre.

Plus récemment, sont apparus les simulations de colonies d'insectes et, plus généralement, les algorithmes en essaim [7, 6]. Dès 1988, des SMA réactifs avaient montré que des dynamiques collectives, tels des vols d'oiseaux, peuvent être simulés de manière réaliste à partir d'agents ayant un comportement individuel très simple [62]. D'une manière générale, ce travail montre que l'activité concurrente d'un ensemble d'agents au comportement individuel simple peut mener à l'émergence de structures de plus grande échelle, telle que la construction d'une termitière, comme suggéré dans [45]. On constate alors que la structure en cours de construction à grande échelle rétro-agit sur l'activité des agents *via*, un processus qualifié de stigmergique, notion introduite historiquement par les entomologistes [31]. Les SMA réactifs sont également utilisés en écologie pour la compréhension des éco-systèmes [17]. L'agent  $y$  représente une entité biologique dont le métabolisme est décrit avec une plus ou moins grande précision [18, 32]. L'animation du modèle par des simulations permet d'explorer l'influence de tel ou tel paramètre et de mieux comprendre l'émergence de certaines propriétés globales. Ainsi, [22] montre l'intérêt de considérer que le taux d'ingestion d'un organisme, généralement considéré comme une constante dans ce type de modèles, est bien la résultante d'une activité biologique. Cela permet d'étudier, de mettre à l'épreuve et de valider plus finement le modèle. Ce type de résultats intéresse particulièrement les biologistes modélisateurs.

Des modèles plus complexes associant algorithme génétique et réseau de neurones ont été étudiés à diverses occasions. Ainsi, on a étudié les interactions entre évolution génétique et apprentissage au cours de la vie, en y ajoutant éventuellement une couche culturelle, menant donc à des études sur l'effet Baldwin (voir notamment [14, 56, 26, 55, 51, 52, 47, 28, 20]). Algorithmes génétiques et réseaux de neurones ont également été utilisés pour engendrer des architec-

tures de contrôle pour des robots (voir notamment [27, 44]).

Sur un plan purement phénoménologique, dès 1950, G. Walter a montré qu'un robot contrôlé par quelques boucles de rétro-actions simples peut exhiber un comportement d'une complexité frappante [78]. Par la suite, Brooks et d'autres ont également travaillé sur cette piste menant à la notion d'architecture de subsomption [9, 10]. Ces robots sont très réactifs (aucune représentation n'est utilisée) et exhibent des comportements rappelant ceux d'insectes.

Pour une partie d'entre-eux, les travaux sur les animats ont pour objet la modélisation du vivant (voir [34] pour un état récent de ces recherches). Une grande part des travaux menés dans cette perspective ont clairement pour objectif de s'inspirer du vivant pour synthétiser des artefacts adaptatifs. Ce point de vue peut être complémentaire d'une recherche spécifiquement dédiée à la compréhension du comportement animal en tant que tel. Notons également que ces travaux mènent à un point de vue renouvelé sur la notion de système cognitif [35].

Explicitement issus de travaux sur la modélisation du comportement animal ont été développées les méthodes dites à différence temporelle (*temporal difference*, abrégé en TD) depuis [3], puis introduites sous ce nom dans [70]. Ces travaux ont rejoint le courant concernant le contrôle optimal pour donner naissance aux algorithmes de renforcement à la fin des années 1980. L'idée de base de ces méthodes est de faire reposer le choix de l'action à effectuer sur l'écart entre les conséquences attendues de l'émission de cette action et les conséquences effectivement observées ; de manière volontairement intuitive, l'idée est que le comportement d'un animal est plus affecté lorsque les conséquences de ses actions le surprennent que dans le cas où il obtient ce qu'il attendait : c'est le principe de la loi de Rescorla-Wagner [61], importante loi de la dynamique du comportement animal ; c'est ce principe que l'on a cherché à mettre sous la forme d'un modèle formel et qui a mené aux méthodes TD. Ce modèle a été utilisé pour rendre compte du conditionnement classique (ou pavlovien), c'est-à-dire, l'acquisition d'un réflexe [71]. Beaucoup plus riche est le conditionnement dit opérant ou instrumental qui décrit l'adaptation du comportement et les apprentissages effectués au cours de sa vie par un animal [69]. Historiquement, ce type d'apprentissage a été modélisé à l'aide de réseaux de neurones [33]. D'autres méthodes ont été proposées [76]. Il peut également être modélisé à l'aide de méthodes TD [2, 72]. Différents travaux ont été réalisés dans cette voie (voir [42] pour un état de l'art récent). Différents auteurs ont bien compris le potentiel de cette approche [48, 21, 76, 63, 59, 15] : si on était capable de faire faire à un robot ou à un logiciel ce que l'on sait faire un animal et ce que l'on sait lui apprendre à faire (dans un cirque, chien d'aveugle, ...), on aurait le sentiment d'avoir effectué un pas important vers la réalisation d'artefacts autonomes capables de se débrouiller dans un environnement *a priori* inconnu. Hors, les méthodes TD sont des modèles de l'apprentissage animal mis en jeu avec ces animaux.

Enfin, pour clore cette brève revue de l'existant, alors que la plupart des travaux précédemment cités entendent rendre compte d'un comportement acquis (à l'exception notable des algorithmes de renforcement), quelques travaux

mettent l'accent sur l'acquisition elle-même [49, 64]. Considérant le comportement comme étant en perpétuelle adaptation dans un environnement changeant, l'acquisition d'un comportement n'est qu'un moment particulier de cette dynamique. Aussi, si l'on veut reproduire le vivant, il faut définir un artefact dont le comportement est en perpétuelle adaptation ; si on veut qu'il apprenne un comportement particulier qui ne fait pas partie de son répertoire comportemental spontané, il faut modifier son environnement pour que, s'adaptant à ce changement, il émette le comportement voulu. C'est de cette manière que l'on apprend à un animal un comportement non naturel (faire du vélo pour un ours, ...) et c'est également par interaction avec leur milieu que les espèces animales évoluent au cours des générations.

### 3 Modèle fondé l'apprentissage de la différence temporelle

Nous précisons dans cette section ce que sont les algorithmes de renforcement. Pour cela, on définit le problème de l'apprentissage par renforcement que ces algorithmes résolvent. Ensuite, on présente l'un d'eux, le Q-Learning.

#### 3.1 Le problème d'apprentissage par renforcement

Les algorithmes basés sur la notion d'apprentissage de la différence temporelle (TD) résolvent le problème de l'apprentissage par renforcement. Définissons ce problème. Considérons un automate évoluant dans un temps discret. Soient :

- un ensemble d'états  $\mathcal{S}$  décrivant les états possibles de l'automate. Parmi les états, certains peuvent être terminaux ;
- un ensemble d'actions possibles dans l'état  $s$  :  $\mathcal{A}(s), \forall s \in \mathcal{S}$  ;
- une fonction de transition :  $\mathcal{P}_{ss'}^a$ , qui donne la probabilité de passer de l'état  $s$  dans l'état  $s'$  suite à l'émission de l'action  $a$  ;
- une fonction de retour immédiat :  $\mathcal{R}_{ss'}^a$ , qui donne l'espérance de retour immédiat suite au passage de l'état  $s$  dans l'état  $s'$  résultant de l'émission de l'action  $a$ .

Schématiquement, l'automate fonctionne comme suit. À l'instant  $t$ , il se trouve dans un état  $s$  qui synthétise sa perception de l'environnement courant ainsi que son état interne ; il choisit alors une action à effectuer parmi  $\mathcal{A}(s)$  ; l'émission de cette action provoque un changement d'état conformément à  $\mathcal{P}$  et un retour immédiat (une conséquence) conformément à  $\mathcal{R}$ . On passe à l'instant  $t + 1$ .

On définit le retour :  $R = \sum_{t=0}^{t=\infty} \gamma^t r_t$  où  $r_t$  est le retour immédiat obtenu à l'instant  $t$ . Ce retour est donc la somme des retours immédiats obtenus au cours du fonctionnement de l'automate ; le coefficient  $\gamma \in [0, 1]$  (taux de dépréciation

ou *discount rate*) donne une plus ou moins grande importance aux retours à court terme par rapport aux retours à long terme.

On définit une politique (ou stratégie)  $\pi(s, a)$  qui spécifie pour chaque état  $s$  la probabilité pour l'automate d'émettre l'action  $a$ . Le problème d'apprentissage par renforcement consiste à trouver une politique  $\pi^*(s, a)$  qui maximise  $R$ .

Pour préciser l'algorithme résolvant ce problème, on définit :

- la valeur d'un état  $s$  (notée  $V^\pi(s)$ ) est l'espérance de gain après avoir visité l'état  $s$  si l'automate suit la stratégie  $\pi$  :  

$$V^\pi(s) = E_\pi(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, s_t = s) ;$$
- la qualité d'un couple état  $s$ , action  $a$  (notée  $Q^\pi(s, a)$ ) est l'espérance de gain suite à l'émission du comportement  $a$  dans l'état  $s$  si l'automate suit la stratégie  $\pi$  :  $Q^\pi(s, a) = E_\pi(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, s_t = s, a_t = a) ;$

On note  $V^*$  et  $Q^*$  les valeurs optimales de ces quantités qui correspondent à une stratégie optimale  $\pi^*$ . Dans un environnement markovien<sup>2</sup>,  $V^*$  et  $Q^*$  peuvent être calculées par divers algorithmes itératifs. Toujours dans un environnement markovien, une fois  $Q^*$  obtenu,  $\pi^*$  peut en être aisément déduite en adoptant un algorithme glouton : pour chaque état  $s$ , la stratégie optimale consiste à effectuer l'action (ou l'une des actions s'il y en a plusieurs) dont la qualité est maximale dans cet état.

Ainsi, dans le cas où ces 4 données  $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$  sont connues *a priori*, il existe un algorithme qui calcule  $\pi^*$  avec une complexité temporelle proportionnelle au nombre d'états au carré : c'est la méthode de programmation dynamique. Nous ne décrivons par cette méthode ici (voir par exemple [72]). Simplement, nous exprimerons quelques critiques :

- le nombre d'états est généralement très grand ; donc, une complexité proportionnelle au carré de ce nombre devient rapidement impraticable. Exemple : pour un jeu aussi simple que le tic-tac-toe, le nombre d'états est  $|\mathcal{S}| = 3^9$ , ce qui, au carré, donne environ 400 millions ; ne parlons pas du jeu d'échecs et de ses  $\approx 10^{100}$  états différents ;
- il faut disposer d'un modèle parfait de l'environnement : c'est une hypothèse irréaliste pour un problème en vraie grandeur ;
- il faut que les distributions de probabilité décrivant l'environnement soient stationnaires. Cette hypothèse est impossible à satisfaire si l'on s'intéresse à deux agents adaptatifs en interaction (par exemple, deux agents adaptatifs apprenant à jouer à un jeu l'un contre l'autre) : étant adaptatifs,

---

<sup>2</sup>un environnement est markovien si la connaissance de l'état courant permet de choisir l'action optimale à effectuer, donc sans aucune connaissance des états ayant précédé : c'est la négation de la notion d'historique. On peut formuler cette propriété autrement en disant que dans un environnement markovien, la probabilité d'émettre une certaine action ne dépend que de l'état courant et pas des états précédents ; en résumé, l'état courant condense toute l'histoire de l'agent et lui permet de prendre la bonne décision quant à la meilleure action à effectuer dans l'état courant.

leurs stratégies évoluent au cours du temps, modifiant donc la probabilité d'émission d'un comportement donné dans un état donné au cours du temps, ce qui est la négation de la stationnarité de l'environnement ;

- il faut que les états soient définis une fois pour toute, que des états ne puissent être ni ajoutés, ni modifiés, ni supprimés. Par exemple, cette fixité de la définition des états entraîne la fixité de la perception de l'agent : il ne peut pas apprendre de nouveaux stimuli au cours de son existence : cela revient à nier la capacité d'apprentissage de la perception de l'agent ;
- de même, il faut que les actions possibles soient définies une fois pour toutes : à nouveau, c'est nier les capacités d'apprentissage de l'agent que l'on veut adaptatif. De plus, la restriction initiale du répertoire comportemental alliée au développement de ce répertoire au cours de l'existence de l'agent permet de simplifier grandement la tâche d'apprentissage au cours de la vie ; initialement, le petit n'a pas le choix entre attraper un objet, marcher, courir, nager, ... Dans le cas du bébé humain, il ne sait rien faire de tout cela ; il sait juste effectuer des mouvements désordonnés qui, petit à petit, vont se coordonner pour construire et structurer son répertoire comportemental. Celui-ci doit donc pouvoir évoluer ;
- enfin, cette hypothèse n'a aucune validité par rapport au vivant pour les raisons qui viennent d'être rapidement évoquées.

D'autres algorithmes ont été proposés, les méthodes à base de différence temporelle qui ont une certaine pertinence par rapport à la dynamique comportementale du vivant comme on l'a vu plus haut (*cf.* section 2.2). Nécessitant moins d'hypothèses que la programmation dynamique pour leur mise en œuvre, ces méthodes fournissent asymptotiquement  $Q^*$ , donc  $\pi^*$ , pour des environnements markoviens ; expérimentalement, on sait que l'on obtient des résultats intéressants au bout d'un nombre fini d'itérations et que ces méthodes fonctionnent aussi dans des environnements non markoviens. Dans ce cas, on ne dispose d'aucun résultat formel, seulement de preuves expérimentales ; un autre de leurs intérêts est de permettre un apprentissage en continu : ceci est particulièrement intéressant pour les systèmes en constante interaction avec leur environnement : l'algorithme est en constante adaptation par rapport à son milieu, qui peut donc ne pas être stationnaire.

### 3.2 Méthodes à base de différence temporelle

Les méthodes à base de différence temporelle (désormais abrégé en méthode ou algorithme TD) sont des algorithmes itératifs qui, à chaque itération, décident de l'action à effectuer en fonction de leur expérience passée, et corrigent leur estimation de l'intérêt d'effectuer cette action dans cet état (la qualité), en fonction du retour immédiat obtenu et de leur expérience passée. Contrairement à un algorithme d'apprentissage supervisé (réseau de neurones avec rétro-propagation des erreurs par exemple), la meilleure action à effectuer dans un état donné n'est

jamais indiquée à un algorithme TD ; de même, l'algorithme ne reçoit aucune information lui indiquant s'il aurait pu effectuer une meilleure action ; il reçoit simplement un retour immédiat encore appelé renforcement.

Insistons également sur le fait qu'aucun modèle de l'environnement n'est nécessaire pour utiliser un algorithme TD. Une fois définie l'architecture de l'agent (les états et les actions possibles), on le laisse interagir avec son environnement. Petit à petit, il va s'y adapter et apprendre par lui-même l'action qu'il faut effectuer dans une situation donnée. Ces algorithmes font partie des méthodes d'apprentissage automatique [53].

### 3.3 Un exemple d'algorithme de la classe TD : le Q-Learning

Parmi les méthodes TD, nous définissons plus précisément l'algorithme Q-learning qui est utilisé dans le reste de cet article. Cette classe de méthodes en comprend d'autres : Sarsa, et les versions utilisant des traces d'éligibilité Sarsa( $\lambda$ ) et Q( $\lambda$ ). L'exposé de ces méthodes nous mèneraient bien trop loin sans être indispensable à la lecture de la suite. Le lecteur intéressé pourra consulter la littérature [5, 43, 72].

Le Q-Learning a été introduit en 1989 [79]. C'est un algorithme itératif qui apprend la qualité des couples (état, action). Il s'exprime comme indiqué par l'algorithme 1. Nous détaillons brièvement ses différentes parties.

---

#### Algorithme 1 Q-Learning

---

```

Initialiser  $Q(s, a)$  arbitrairement pour tous les couples  $(s, a)$ 
pour toujours faire
  Initialiser  $s$ 
  répéter
    Choisir  $a$  (par une sélection  $\epsilon$ -gloutonne par exemple)
    Observer  $s'$  et  $r$ 
    Modifier  $Q(s, a)$  :  $Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_b Q(s, b) - Q(s, a))$ 
     $s \leftarrow s'$ 
  jusque  $s$  est un état terminal
fin pour

```

---

- l'initialisation des qualités est arbitraire : on peut mettre toutes les qualités à 0, les initialiser avec une valeur aléatoire, ou les initialiser avec des estimations de leur valeur réelle si l'on dispose d'information sur l'environnement ;
- une boucle infinie consiste alors à initialiser l'état initial de l'automate puis à laisser évoluer et s'adapter l'algorithme jusqu'à ce qu'un état terminal soit atteint. Lorsqu'il atteint un état terminal, un épisode est terminé ; l'automate est replacé dans un état initial (pas nécessairement le même que la fois précédente) et un nouvel épisode démarre. Notons que la notion d'état terminal peut être dénuée de sens pour certains environnements ;

- la boucle Répéter parcourt les états d'un épisode :
  - à chaque épisode, étant dans l'état courant  $s$ , une action est sélectionnée ; plusieurs algorithmes de sélection sont possibles : on peut par exemple sélectionner l'action dont la qualité est maximale dans l'état courant avec une probabilité  $1 - \epsilon$  et prendre une action au hasard avec une probabilité  $\epsilon$  (sélection dite  $\epsilon$ -glouton). Partant de valeurs  $Q$  initialisées arbitrairement, il est rationnel de ne pas porter beaucoup d'intérêt à ces valeurs pour sélectionner l'action à effectuer en début d'épisode et explorer l'espace des actions ; ensuite, les valeurs  $Q$  tendant petit à petit vers  $Q^*$ , le choix de l'action peut se baser de plus en plus sur celles-ci, menant à une exploitation de la recherche déjà effectuée. Il est donc intéressant de faire décroître  $\epsilon$  au cours du temps. On peut également associer une probabilité de sélection à chaque action  $a$  qui soit fonction des valeurs  $Q(s, a)$ . Le problème est alors de trouver la distribution de probabilités qui assure une probabilité de sélection judicieuse à chaque action en fonction de sa qualité  $Q(s, a)$ . On choisit généralement une probabilité de sélection proportionnelle à  $Q(s, a)$ , ou à  $e^{Q(s, a)}$  ;
  - suite à l'émission de l'action, on observe l'état  $s'$  dans lequel l'algorithme va passer et le retour immédiat  $r$ . En fonction de ces données, la qualité du couple  $s, a$  est mise à jour. L'algorithme venant d'échantillonner les conséquences de l'émission de l'action  $a$  dans l'état  $s$ , cette mise à jour combine ce retour immédiat avec les retours qui sont espérés dans la suite (le terme  $\max_b Q(s, b)$  de l'équation de mise à jour de  $Q(s, a)$ ) en les pondérant par le taux de dépréciation  $\gamma$ . Un paramètre, le taux d'apprentissage  $\alpha$ , équilibre l'influence de la valeur courante de  $Q(s, a)$  avec sa nouvelle estimation ; si  $\alpha = 0$ , aucun apprentissage n'a lieu ; si  $\alpha = 1$ , la valeur courante de  $Q(s, a)$  n'est pas utilisée et seuls comptent les retours futurs estimés et le retour immédiat ; en général, on prend une valeur intermédiaire entre 0 et 1 ;
  - après la mise à jour de  $Q(s, a)$ , l'algorithme passe effectivement dans le nouvel état  $s'$  et une nouvelle itération est effectuée, à moins qu'un état terminal n'ait été atteint provoquant la sortie du répéter.

Watkins [79] a prouvé formellement que le Q-learning converge asymptotiquement vers la politique optimale dans un environnement markovien, du moment que chaque paire (état, action) est visitée une infinité de fois et que  $\alpha$  décroît adéquatement au cours du temps, tendant asymptotiquement vers 0. De même, la convergence de Sarsa a été prouvée dans le même cadre. En revanche, on n'a pas de preuves d'optimalité pour les algorithmes utilisant des traces d'éligibilité Sarsa( $\lambda$ ) et Q( $\lambda$ ). Cependant, d'une manière générale, on a constaté expérimentalement que ces algorithmes fonctionnent de manière très convenable, même en dehors du domaine où ils ont été prouvés : comportement

non asymptotique, environnement non markovien, environnement non stationnaire.

Ces algorithmes constituent donc des outils puissants pour la résolution de problèmes pratiquement difficiles. Ils ont été appliqués à la résolution de différents problèmes réels, tels que la gestion de groupes d'ascenseurs dans un immeuble.

Notons que par rapport aux critiques effectuées sur la programmation dynamique un peu plus haut, diverses adaptations du schéma d'algorithme Q-Learning sont possibles : par exemple, l'ensemble des états peut varier au cours du temps, des états étant créés, d'autres fusionnant, d'autres encore disparaissant. De même, l'environnement peut ne pas être stationnaire : n'ayant pas à construire un modèle de l'environnement, il peut évoluer librement sans que cela complexifie l'algorithme.

### 3.4 Lien entre TD et comportement animal

TD et Q-learning en particulier reposent bien sur l'idée de la sélection du comportement par ses conséquences : une action suivie d'un retour meilleur que les autres actions possibles dans le même état voit sa probabilité d'émission croître avec sa qualité. Ce point a été argumenté à plusieurs reprises [2, 72]. TD présente donc un algorithme pertinent et donc un moyen d'explorer la dynamique comportementale de l'animal. C'est à cette exploration qu'est dédié le travail décrit dans la section suivante. Notons cependant que TD peut être critiqué comme modèle de l'apprentissage du comportement ; nous y reviendrons dans la discussion finale.

L'ensemble des actions  $\mathcal{A}$  constitue le répertoire comportemental. À chaque action est associée une probabilité d'émission dans un état donné au travers de la qualité du couple (état, action). Au cours de l'apprentissage, ces probabilités évoluent au gré des conséquences de leurs émissions. Des séquences d'actions sont sélectionnées et acquises rendant naturels certains comportements initialement très peu probables, provoquant donc l'apparition de comportements nouveaux ; le qualificatif nouveau est donc malapproprié : il s'agit de comportements dont la probabilité d'émission, initialement très faible, augmente au gré de l'histoire de l'organisme et de son développement ; rappelons que ce développement est lui-même sujet, dans une certaine mesure, aux interactions entre l'organisme et son milieu ; certaines stimulations doivent être reçues pour que certaines aptitudes, certains sens, se développent, même si elles sont codées génétiquement.

## 4 Un exemple de modèle du comportement animal : MAABAC

MAABAC est une classe d'architectures multi-agents de renforcement modélisant un organisme multi-segmenté. Le contrôle de chaque agent repose exclusivement sur un algorithme TD. Aucun superviseur ne contrôle ni ne coordonne

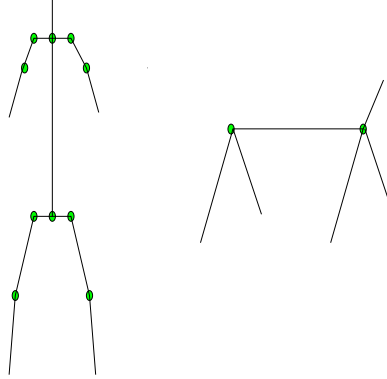


Figure 1: L'allure générale de deux représentants de la classe d'automates MAABAC donnés à titre d'illustration. On distingue les os (traits) et les articulations (disques). L'automate dont il est question dans le texte correspond à la représentation de gauche.

les activités des agents. L'objectif de cette étude est d'une part d'explorer les possibilités de ce type d'architectures, d'autre part de servir de modèles pour l'analyse du comportement animal.

MAABAC a été introduit par [19], complété dans [12]. Classe générale d'architectures, MAABAC peut donc prendre de très nombreuses formes. Nous en indiquons deux à la fig. 1 : l'une qui ressemble à un humanoïde, l'autre à un quadripède. Nous travaillons actuellement sur la réalisation d'un humanoïde complet. L'objectif est de lui faire apprendre différents comportements et d'étudier ces apprentissages : placer sa main, puis ses mains, dans une certaine zone de l'espace pour y attraper un objet ; lorsque cet objet est trop éloigné pour l'attraper, MAABAC devra apprendre à se déplacer (marcher, ramper, courir, ...). Nous souhaitons également qu'il soit capable d'apprendre à utiliser des objets si cela lui est nécessaire : s'asseoir sur une chaise, ramer dans une barque, ... Nous partons de l'hypothèse qu'un organisme vivant apprend à émettre des comportements en fonction des conséquences passées de leur émission, donc en fonction de l'environnement dans lequel il se trouve. Outre l'apprentissage de séquences de comportements, nous nous intéressons particulièrement à la phase d'apprentissage en elle-même : MAABAC se veut avant tout un modèle du vivant et ce n'est pas la réalisation d'une certaine performance qui est attendue, mais bien son acquisition qui nous intéresse.

MAABAC est un assemblage d'os, de muscles et d'articulations (virtuels). Les muscles contrôlent les positions et les rotations des articulations. Dans un modèle 2D, chaque articulation est contrôlée par deux muscles, l'un agoniste, l'autre antagoniste : chacun tire dans son sens et l'action des deux place l'articulation dans une certaine position. Dans un modèle 3D, deux à quatre muscles contrôlent une articulation, selon ses degrés de liberté : mouvement dans un plan (comme le coude) ou dans deux plans (comme l'épaule). Le mus-

cle est donc l'agent actif de MAABAC ; l'hypothèse est que le muscle apprend son comportement en respectant le principe de sélection du comportement par ses conséquences. D'un point de vue physiologique, ce que nous appelons ici un agent muscle est donc constitué par les moto-neurones contrôlant un véritable muscle. Ces moto-neurones contrôlent la contraction du muscle et ont leur comportement qui s'adapte au cours du temps. Élément central de MAABAC, la section 4.1 décrit l'agent muscle. Ensuite, la section 4.2 décrit l'ensemble de l'architecture logicielle de MAABAC. La section 4.3 présente des expérimentations réalisées à ce jour.

## 4.1 MAABAC : un système d'apprentissage par renforcement

Chaque agent musculaire est contrôlé par un algorithme Q-Learning. En absence de perception, l'état d'un agent est le niveau de contraction du muscle, caractérisé par un entier compris entre 0 et  $N$ . Trois actions sont possibles : augmenter d'une unité le niveau de contraction du muscle, relâcher d'une unité la contraction musculaire, la laisser inchangée. Les retours sont : la récompense éventuellement perçue lorsqu'une tâche a été correctement réalisée ; le coût énergétique du à la contraction du muscle  $-\kappa \frac{c}{N}$  où  $\kappa$  est une constante  $\in [0, 1]$  et  $c$  est le taux de contraction du muscle. Dans le cas où MAABAC perçoit son environnement, cette perception est combinée avec le niveau de contraction pour former l'état de MAABAC.

Pour un agent musculaire, la réception d'un renforcement ne dépend pas uniquement de son propre comportement. C'est le comportement coordonné de l'ensemble des agents qui entraîne la récompense. Par contre, l'action de chaque muscle provoque un coût énergétique propre à chaque agent. Le problème que doit résoudre chaque agent n'est donc pas markovien : l'état de contraction ne permet pas à chaque agent de déterminer l'action à réaliser pour maximiser son espérance de gain ; c'est bien la conjonction des états de contraction des différents agents qui détermine le retour ; or, chaque agent ne dispose que de son propre état de contraction, pas des états de contraction des autres agents. Nous allons cependant voir dans les expérimentations que MAABAC, et donc chacun de ses agents musculaires, parvient petit à petit à s'approcher d'une politique optimale.

## 4.2 MAABAC : une architecture multi-agents

Au niveau de l'architecture logicielle (*cf.* fig. 2), MAABAC est constitué d'un ensemble d'agents musculaires. Ces agents agissent de manière décentralisée. Chaque agent musculaire étant un Q-Learning, une itération de MAABAC consiste en une itération de chacun des Q-Learning. À chaque itération de MAABAC, sa position dans l'espace est déterminée en fonction des états de contraction des muscles. Cette position entraîne des conséquences : lorsque la tâche assignée à MAABAC est réalisée, une conséquence est délivrée. De même, l'activité de MAABAC peut entraîner des modifications de l'environnement, par

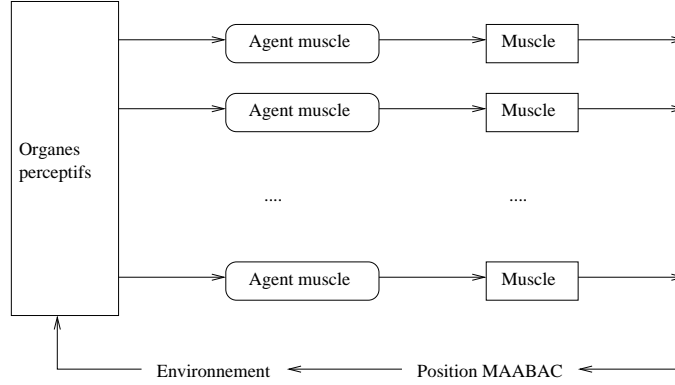


Figure 2: L'architecture logicielle générale de MAABAC. Voir le texte.

exemple s'il est capable de bouger un objet ou de consommer une ressource. Ces modifications de l'environnement sont perçues par les organes perceptifs de MAABAC : ces organes peuvent ne percevoir que la présence ou non d'une conséquence ou être plus évolués (perception visuelle, tactile, ...). Dans ce qui suit, nous n'avons pas muni MAABAC d'une quelconque perception excepté la perception de la récompense reçue quand la tâche attendue a été réalisée. La structure multi-agents a été choisie pour pouvoir facilement compléter MAABAC et donc le développer incrémentalement.

### 4.3 Expérimentations

#### 4.3.1 Acquisition d'un comportement

Nous expérimentons tout d'abord le mouvement d'atteinte du bras de MAABAC. On ne s'intéresse donc ici qu'à un seul bras composé de deux os et de deux articulations : l'épaule fixe et le coude (*cf.* fig. 3). L'extrémité libre du bras constitue sa main. Une zone de l'espace est définie comme la zone cible dans laquelle MAABAC doit mettre sa main. MAABAC n'a aucune information sur la position de cette zone. Initialement, il ne sait même pas qu'il doit mettre sa main quelque part. À chaque action, le coût énergétique est perçu comme un retour négatif ; s'il parvient à mettre sa main dans la zone cible, un retour valant  $+1$  lui est transmis, soit  $1 - \kappa \frac{c}{N}$  pour chacun des muscles. Quand il aura enfin mis sa main dans la zone cible et reçu ce renforcement positif, son bras est remis dans sa position initiale. Il cherchera ensuite à replacer sa main dans cette zone. Notons qu'en parallèle à ce travail de modélisation, des expérimentations ont été réalisées sur de très jeunes (quelques jours) bébés humains pour y confronter les résultats des simulations.

Précisément, les angles de flexion des deux articulations sont :

- pour l'épaule :  $\theta_0 = \frac{\pi}{2} \frac{c_0 - c_1}{N} + \frac{\pi}{2}$  ;
- pour le coude :  $\theta_1 = \frac{\pi}{2} \frac{c_2 - c_3}{N} + \frac{\pi}{2} - \theta_0$ .

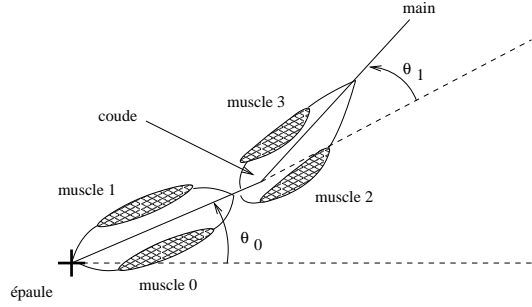


Figure 3: Une représentation schématique du bras de MAABAC en 2D. Voir le texte.

où  $c_0$ ,  $c_1$ ,  $c_2$  et  $c_3$  représentent les états de contraction musculaire des 4 muscles  $m_0$ ,  $m_1$ ,  $m_2$  et  $m_3$ .

Pour être précis quant à la description du bras de MAABAC, son bras est constitué de deux segments de longueur unitaire. Le nombre d'états de contraction possible  $N$  a été fixé à 50.  $\theta_0$  et  $\theta_1$  sont compris entre 0 et  $\pi$  radians : cela représente grossièrement les amplitudes angulaires de nos articulations. Dans les Q-Learning, le taux de dépréciation  $\gamma$  est fixé à 0.9 tandis que le taux d'apprentissage  $\alpha$  est fixé à 0.5. La probabilité de sélectionner une action est proportionnelle à sa qualité par rapport aux autres actions possibles dans l'état courant. En aucun cas, nous n'avons cherché un jeu de paramètres optimal ou la méthode de sélection optimale. Aussi, les performances décrites dans la suite ne sont-elles probablement pas les performances optimales que l'on puisse obtenir avec cette d'architecture.

**Procédure** La procédure est indiquée par l'algorithme 2. Elle consiste simplement à placer le bras dans une position initiale puis à le laisser atteindre de lui-même la zone cible. Lorsqu'il l'a atteinte et y est demeuré pendant quelques itérations, on remet le bras dans la position initiale et on recommence. Il s'agit donc d'une tâche épisodique. On appelle cette séquence d'actions un mouvement. On effectue ce mouvement NT fois. Pour chaque atteinte, on mesure le nombre d'itérations effectuées par MAABAC.

**Résultat** L'apprentissage du mouvement d'atteinte est visualisé à la fig. 4. On observe que l'acquisition du comportement peut être décrite en cinq phases :

- MAABAC présente un mouvement non coordonné qui paraît désordonné (fig. 4, 1<sup>re</sup> vignette à gauche) ;
- l'extrémité du bras passe accidentellement dans la zone de renforcement sans y rester (fig. 4, 2<sup>e</sup> vignette depuis la gauche) ;

---

**Algorithme 2** Procédure d'apprentissage d'un mouvement d'atteinte

---

Initialiser les qualités arbitrairement pour chacun des agents muscles  
Déterminer un état de contraction initial pour chacun des muscles

**pour** tentative  $\in [1, NT]$  **faire**

    Initialiser l'état de chacun des agents muscles avec son état de contraction initial

**répéter**

        Sélectionner l'action pour chacun des agents muscles (en suivant un Q-learning)

        Calculer la position de la main de MAABAC et en déduire le retour immédiat :

$$r = \begin{cases} 0 & \text{si la main est en dehors de la zone cible,} \\ 1 & \text{si elle est à l'intérieur} \end{cases}$$

        Donner le retour ( $r +$  coût énergétique) à chacun des muscles

**jusque** la main demeure à l'intérieur de la zone cible un certain laps de temps

**fin pour**

---

- lorsque l'extrémité du bras est suffisamment proche de la zone de renforcement, le système peut l'atteindre (fig. 4, 2<sup>e</sup> vignette depuis la gauche) ;
- depuis le point de départ, le système est capable d'atteindre la zone de renforcement mais le mouvement comporte des irrégularités et des retours en arrière (fig. 4, 3<sup>e</sup> vignette depuis la gauche) ;
- finalement, le comportement d'atteinte est exécuté de façon coordonnée et relativement directe (fig. 4, vignette à droite).

De manière plus précise, on peut représenter le nombre d'itérations nécessaires pour effectuer une atteinte (*cf.* fig. 5). Important en début d'apprentissage (de 500 à 1500 environ), ce nombre décroît ensuite après une cinquantaine

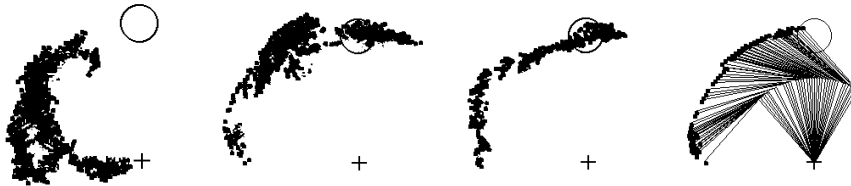


Figure 4: Différentes phases de l'acquisition d'un mouvement d'atteinte par le bras de MAABAC. Sur les 3 vignettes de gauche, seules les différentes positions de la main sont indiquées. La croix indique la position de l'épaule et le disque représente la position de la zone cible où MAABAC doit placer sa main. À droite, on a représenté les différentes positions du bras au cours d'un mouvement d'atteinte complet, une fois celui-ci acquis.

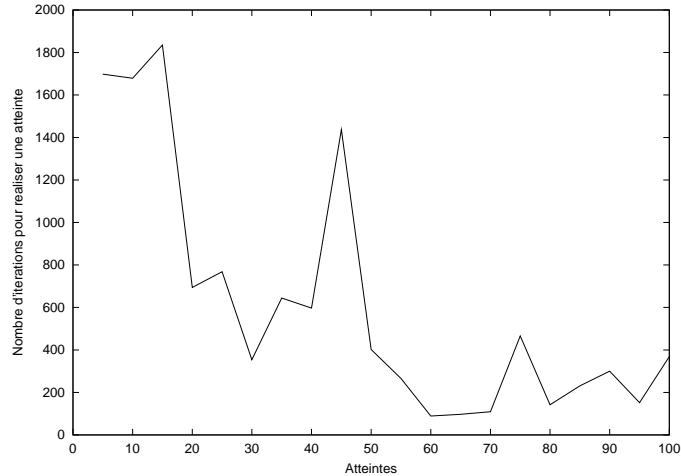


Figure 5: Acquisition du comportement d’atteinte par le bras de MAABAC. Le nombre d’itérations pour effectuer un mouvement d’atteinte depuis une position initiale du bras jusqu’à la cible est indiqué. Au fil des mouvements, ce nombre d’itérations diminue.

d’atteintes. Le nombre d’itérations tourne alors autour d’une centaine, ce qui correspond à un mouvement relativement direct vers la cible.

**Discussion** Grosso modo, la courbe d’apprentissage obtenue ressemble à celle qui est obtenue sur de jeunes bébés. (Une comparaison stricte de la fig. 5 avec ce qui est obtenu avec le bébé est impossible.) De manière plus significative, certaines observations effectuées avec le bébé sont par ailleurs effectuées sur MAABAC également :

- le bébé, et MAABAC, ont tendance à contracter les muscles agonistes et antagonistes contrôlant une articulation en même temps ce qui provoque une grande dépense d’énergie et des mouvements imprécis et saccadés. Petit à petit, au fil des expériences, le nombre de ces co-contractions musculaires diminue jusqu’à disparaître chez le bébé et MAABAC ;
- le bébé et MAABAC montrent une amplitude de mouvements différentes à l’épaule et au coude [73]. Tous deux exhibent une plus grande amplitude de mouvements à l’épaule qu’au niveau du coude.

Pour chaque agent, les actions possibles, dans un état donné, sont au nombre de trois. Le mouvement d’atteinte complet qui a été appris constitue donc une séquence comportementale plutôt qu’un seul comportement. Ainsi, dans cet exemple, MAABAC a appris une séquence d’environ 50 actions coordonnées entre ses 4 agents musculaires. Ayant vérifié qu’une séquence comportementale peut être acquise par MAABAC, nous vérifions maintenant que différents

phénomènes importants de la dynamique comportementale sont présents : la capacité d'apprendre des mouvements d'atteinte pour différentes zones cibles, ces apprentissages se combinant plutôt que se remplaçant l'un l'autre ; la possibilité d'éteindre un comportement ; la possibilité de façonner un comportement difficile à émettre naturellement. Nous étudions ces différents comportements dans les sections suivantes.

### 4.3.2 Extinction et ré-apprentissage

**Procédure** L'environnement ne favorisant plus l'émission d'un comportement donné, l'extinction consiste à faire en sorte que son émission diminue. Cependant, après extinction, si les conditions de l'environnement font que ce comportement redevient utile, son émission va recommencer ensuite facilement.

Après cette extinction, le comportement n'est plus observé mais la question se pose de savoir dans quelle mesure l'apprentissage préalable est encore disponible. Dans le cas de l'apprentissage animal, il ne s'agit pas d'une simple ré-initialisation du comportement. Lorsqu'on recommence la procédure d'apprentissage, elle se déroule nettement plus rapidement que la première fois.

---

**Algorithme 3** Procédure d'extinction puis ré-acquisition d'un comportement

---

```
// Acquisition initiale du comportement
Acquérir un comportement en récompensant son émission
// Extinction
pour  $i \in [1, NE]$  faire
    Remettre le bras dans sa position initiale
    Laisser le bras se positionner dans la zone cible sans donner de récompense
fin pour
// Ré-acquisition
pour  $i \in [1, NA]$  faire
    Remettre le bras dans sa position initiale
    Laisser le bras se déplacer et le récompenser lorsqu'il met sa main dans la
    zone cible
fin pour
```

---

**Simulation** Suite à la phase d'apprentissage, nous cessons donc de distribuer le renforcement lorsque l'extrémité du bras entre dans la zone de renforcement. Cela dit, lorsque ce mouvement est observé, on replace le bras de MAABAC dans sa position initiale. Il est donc possible d'enregistrer le nombre d'itérations nécessaires pour réaliser le comportement qui avait été acquis (placer la main dans la zone cible) et, de cette manière, quantifier la disparition du comportement.

**Résultats** La figure 6 montre l'évolution du comportement dans la procédure d'extinction. Après une trentaine d'atteintes n'apportant plus de récompense,

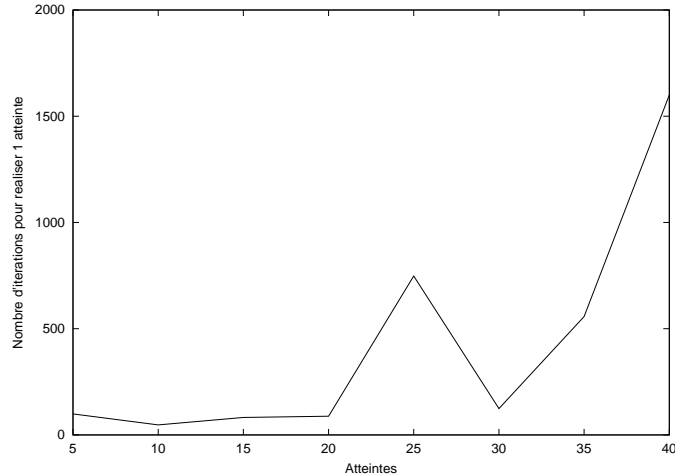


Figure 6: Phase d'extinction du comportement. Voir le texte.

on observe une augmentation du temps nécessaire pour l'atteinte. Au fil des atteintes non récompensées, le temps d'atteinte s'allonge ainsi jusqu'à ce qu'il était avant toute acquisition de comportement (*cf.* fig. 5) : le placement de la main dans la zone cible est redevenu aléatoire.

Après cette extinction, MAABAC effectue, en apparence, des mouvements désordonnés et exploratoires. À ce moment-là, la récompense est à nouveau donnée en cas de positionnement correct de la main dans la zone cible. On constate alors que MAABAC ré-apprend à placer sa main dans la zone cible et ce second apprentissage est beaucoup moins long que le premier (*cf.* fig. 7) : MAABAC n'avait donc pas complètement oublié le mouvement appris mais l'avait simplement délaissé.

### 4.3.3 Le façonnage

**Procédure** Comme on l'a indiqué dans l'introduction, le façonnage est un mode opératoire extrêmement puissant et important en dynamique comportementale. L'objectif est de faire acquérir un comportement qui est initialement très difficile à émettre, c'est-à-dire dont la probabilité d'émission spontanée est extrêmement faible. Une procédure d'apprentissage consistant simplement à récompenser le comportement voulu lorsqu'il est émis est inopérante dans ce cas, la probabilité d'émission spontanée étant beaucoup trop faible. Aussi, pour réaliser cet apprentissage, on imagine une séquence de comportements qui vont être acquis les uns après les autres pour s'approcher petit à petit du comportement visé. Cet apprentissage constitue le façonnage du comportement. Par façonnage, on peut en principe faire acquérir n'importe quel comportement par n'importe quel organisme vivant du moment qu'il lui est physiquement possible.

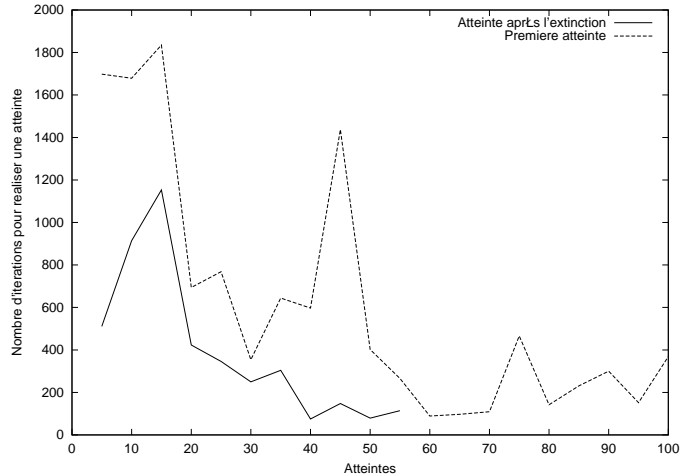


Figure 7: Ré-acquisition du comportement. La courbe d'apprentissage des premières atteintes avant l'extinction est en pointillés. La courbe de ré-apprentissage après l'extinction est en trait plein. On constate que le ré-apprentissage est beaucoup plus rapide, signe que les premiers apprentissages n'avaient pas été perdus lors de l'extinction, mais simplement mis de côté.

**Simulation** Dans le cas de MAABAC, un comportement peu probable est choisi. Ce comportement correspond à une configuration extrême qui requiert une extension maximale du bras. Pour cela, il faut que tous les muscles de MAABAC soient simultanément dans un état précis. En l'absence de renforcement, cette configuration s'observe spontanément après environ 20 000 itérations en moyenne. Cependant, elle ne peut être apprise de cette manière, le coût énergétique étant très important dans cette position du bras.

Pour faire apprendre ce comportement à MAABAC, nous avons placé le centre de la zone cible à l'extrémité du bras en position étendue et nous avons fait varier son rayon. Initialement de très grande taille, le rayon a été diminué au fur et à mesure de l'apprentissage jusqu'à reprendre sa taille habituelle dans nos expériences. La procédure de façonnage a donc consisté à effectuer la procédure décrite par l'algorithme 4.

---

**Algorithme 4** Procédure de façonnage du comportement d'atteinte

---

**pour** le rayon de la zone cible  $r$  décroissant de  $R_{\max}$  à  $R_0$  **faire**  
  **pour** essai  $e \in [1, NE]$  **faire**  
    Apprendre à placer la main dans la zone cible de rayon  $r$   
  **fin pour**  
**fin pour**

---

**Résultats** Le rayon de la zone cible a été diminué régulièrement 15 fois pour reprendre sa taille habituelle. L'ensemble de la procédure a demandé 11 000 itérations, nombre à comparer aux 20 000 itérations nécessaires pour que MAABAC place spontanément sa main dans la zone. De plus, après cet apprentissage, MAABAC place ensuite sa main en 40 itérations dans la zone, c'est-à-dire, aussi vite dans cette zone que dans les autres zones apprises précédemment. L'efficacité de la procédure de façonnage est donc flagrante sur cet exemple.

#### 4.3.4 Généralisation

**Procédure** La généralisation d'un apprentissage désigne un processus par lequel tout ou partie de cet apprentissage est disponible dans un contexte différent de celui qui a prévalu lors de l'apprentissage. Pour mettre en évidence ce processus, on modifie les conditions dans lesquelles la récompense est donnée.

**Simulation** Après apprentissage de l'atteinte d'une zone depuis une position donnée, le système doit retrouver cette zone depuis des positions de départ plus ou moins éloignées. Pour cela, on commence par une phase d'acquisition que l'on considère comme réussie si trois atteintes nécessitant moins de 40 itérations sont réalisées successivement. Ensuite, on modifie la position initiale du bras et l'on enregistre le nombre d'itérations nécessaires à l'atteinte de la zone cible.

**Résultats** La figure 8 représente le nombre d'itérations nécessaires pour atteindre la cible à partir de position initiale décalées par rapport à la position initiale utilisée pendant l'apprentissage. On constate que pour des positions initiales écartés de moins de  $40^\circ$ , le nombre d'itérations pour atteindre la zone cible reste faible.

#### 4.3.5 D'autres comportements

D'autres expériences ont été réalisées avec le bras de MAABAC :

- apprentissage de l'atteinte dans différentes positions de la zone cible : chaque nouvelle position de la cible se compose avec les positions déjà apprises précédemment ;
- suivi d'une cible en mouvement : une fois une position de la zone cible atteinte, celle-ci peut être imprimée d'un mouvement, pas trop rapide, et MAABAC suit la cible ;
- comportements d'échappement et d'évitement : classiques dans l'étude du comportement animal, ces comportements consistent à fuir des conditions inconfortables de l'environnement. Ainsi, une zone est définie comme zone d'échappement en lui associant une conséquence négative ; quand il y place sa main, MAABAC recevant une conséquence négative, tente de la retirer ; petit à petit, MAABAC apprend à ne plus placer sa main dans cette zone

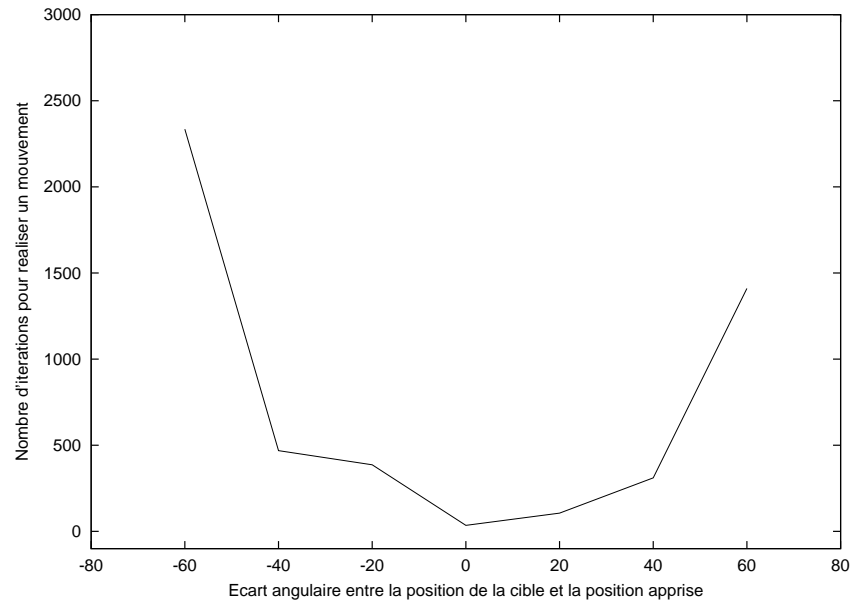


Figure 8: Courbe de généralisation de l'apprentissage d'une zone cible pour différentes positions initiales du bras de MAABAC. En abscisse, on trouve l'écart angulaire entre la position initiale testée et la position initiale utilisée durant l'apprentissage. On voit que tant que cet écart est inférieur à  $40^\circ$ , la zone cible demeure atteinte rapidement.

et à la contourner si elle se trouve sur la trajectoire qui amènerait sa main directement dans la zone cible et montre un comportement d'évitement ;

- généralisation de plusieurs positions cibles : muni d'un organe de perception visuel ultra simple, MAABAC est capable d'apprendre plusieurs positions de la zone cible réparties dans l'espace. Une fois quelques positions apprises, il est capable d'atteindre des positions intermédiaires en un nombre d'itérations faible ;
- le nombre d'états de contraction limite la finesse des mouvements de MAABAC ; par ailleurs, plus ce nombre est élevé, plus le temps d'acquisition d'un mouvement s'accroît. Pour pallier ce problème, il est possible d'apprendre un mouvement avec un nombre relativement faible d'états de contraction et d'augmenter ensuite  $N$  en scindant des états de contraction en deux. Un mouvement précis peut alors être acquis beaucoup plus rapidement. De plus, cette approche met l'accent sur le développement du comportement et l'acquisition de mouvement précis chez le bébé.

#### 4.4 Au-delà du bras de MAABAC

Après l'expérimentation du mouvement d'atteinte du bras de MAABAC, nous avons expérimenté le mouvement d'atteinte de deux bras puis du torse et deux bras. Le deuxième bras possède exactement la même architecture que le premier qui a été décrit. Quant au torse, il peut tourner sur lui-même ; pour cela, il est mu par deux muscles, l'un qui le fait tourner de gauche vers la droite, le second de la droite vers la gauche. En 2D, le torse et les deux bras sont donc constitués de 10 muscles. À nouveau, les muscles sont indépendants les uns des autres ; chacun reçoit la récompense lorsque la tâche est réussie et sa propre dépense énergétique.

Les simulations ont été refaites sur ces deux nouvelles architectures. Dans la suite, nous nous concentrons sur le torse équipé de deux bras.

Le comportement récompensé consiste à placer les deux mains dans la zone cible. Pour cela, MAABAC apprend à orienter son torse de manière à minimiser la dépense énergétique de ses muscles. Acquisition, extinction, façonnage et généralisation sont expérimentés et observés. Le point le plus important concerne l'évolution des temps d'apprentissage quand on passe d'un simple bras au torse avec deux bras. Dans le premier cas, 4 agents sont en jeu ; dans le deuxième, 10 agents doivent coordonner leurs actions, sans aucune supervision. Alors que l'on s'attendrait à ce que le temps d'apprentissage croisse avec le nombre d'agents, on constate exactement le contraire : le torse et ses deux bras acquièrent le comportement d'atteinte en utilisant environ 3 fois moins d'itérations que le bras seul. Le torse et les deux bras étant composés de 2,5 fois de muscles que le bras seul, la durée en temps réel de l'apprentissage demeure donc à peu près constant, voire diminue un peu. Cette observation nous fait penser que la complexification de MAABAC ne va peut être pas s'accompagner d'un fort accroissement des temps d'apprentissage.

## 5 Discussion

Dans cet article, nous nous sommes intéressés à la modélisation du vivant en mettant l'accent sur la modélisation de la dynamique comportementale de l'animal. Pour cela, nous nous sommes appuyés sur la loi de la sélection du comportement par ses conséquences qui peut se traduire sous la forme des algorithmes de renforcement à base de différence temporelle. Nous avons introduit la classe d'automates MAABAC, architectures multi-segmentées dont les éléments actifs, les muscles contrôlant les articulations, sont individuellement contrôlé par un algorithme Q-Learning. Pour que MAABAC effectue un mouvement, il faut donc que ses agents musculaires auto-organisent leurs comportements. Nous avons montré et discuté différentes simulations réalisées à ce jour. Nous insistons sur le fait que nous ne souhaitons pas construire un automate optimal qui résolve un problème particulier ; au contraire, nous voulons construire un automate dont le comportement s'appuie exclusivement sur le principe de la sélection du comportement par ses conséquences et voir si ce principe permet bien de rendre compte de la dynamique comportementale. Dans le présent travail, nous avons montré qu'une architecture multi-agents non supervisée dans laquelle les agents sélectionnent l'action à effectuer en utilisant un algorithme de renforcement est capable de s'auto-organiser pour résoudre des tâches et exhibe naturellement des propriétés classiques dans l'étude du comportement animal : composition de l'apprentissage de plusieurs tâches, apprentissage de séquence comportementale, extinction, façonnage, généralisation, échappement, évitement. Indiquons ici comment notre travail se distingue nettement des autres travaux concernant la modélisation du mouvement d'atteinte du bras. Notre modèle se place à un niveau fonctionnel : l'hypothèse est que les moto-neurones contrôlant l'action des muscles sélectionnent le comportement à émettre en fonction des conséquences de son émission dans le passé. Aucune hypothèse n'est donc faite au niveau mécanique ni neurophysiologique. Le bras du robot COG développé au MIT repose sur un modèle mécanique à base de ressorts et de mouvements primitifs innés [80]. Le bras du robot Darwin III repose sur une modélisation très fine du fonctionnement des neurones [60]. Dans le cas de COG, l'objectif est que le robot soit capable de placer son bras dans une position voulue ; l'accent n'est pas mis sur la manière dont est acquise cette aptitude. Bien entendu, à côté des modélisations plus ou moins réalistes de la dynamique du bras, on trouve les approches purement mathématiques consistant à calculer la trajectoire à faire suivre au bras pour effectuer le mouvement optimal : cette approche, tout à fait adaptée au contrôle de robots dans l'industrie, n'a strictement rien à voir avec celle à laquelle nous nous intéressons ici.

À la suite de ce travail, de nombreuses pistes demeurent à explorer. Concernant MAABAC lui-même, nous voulons poursuivre son développement avec l'ajout de jambes et l'acquisition de déplacement (ramper, marcher, courir, ..., porter un objet, atteindre un objet, ...) ; nous souhaitons le plonger dans un monde physique avec gravité, friction, frottements, ... de manière à ce qu'il puisse apprendre à nager par exemple. Certes, cela nécessite des modélisations

du milieu physique difficiles. Nous voulons également que MAABAC apprenne à se servir d'objets : monter un escalier, utiliser des rames dans une barque pour se déplacer, apprendre à faire du vélo, ...

Sur un plan moins technique, l'utilisation même du Q-Learning dans MAABAC doit être débattue. Le Q-Learning a constitué une première étape et désormais nous utiliserons une variante à base de traces d'éligibilité qui, au lieu de mettre à jour uniquement la qualité du dernier couple (état, action) visité, met à jour les qualités des couples (état, action) visités récemment. Cet algorithme est plus pertinent dans le modèle et devrait accélérer l'apprentissage. Au-delà de ce point, la question de la validité même de l'utilisation de ces algorithmes fonctionnant en temps discret peut être soulevée. Très peu développée jusqu'alors, la perception devra être attaquée et incluse dans MAABAC. Nous n'avons pas immédiatement inclus un organe visuel dans MAABAC pour éviter que la vision résolve, du moins simplifie, les problèmes à la place des agents musculaires. Nous voulions vérifier qu'un ensemble non supervisé d'agents peut auto-organiser son comportement et accomplir des tâches décrites dans le corps de l'article. Nous pouvons maintenant nous intéresser plus précisément à la perception. La perception étant une capacité qui évolue au fil des expériences, la notion d'états au cœur des algorithmes de renforcement devrait se complexifier pour prendre en compte de nouveaux stimuli. De même, comme il a été dit dans le corps de l'article, l'acquisition de nouvelles séquences comportementales peut correspondre à la création de nouvelles actions, et à la modification de l'ensemble des états de l'algorithme. Ces différents points nous mènent tout droit à la nécessité d'avoir une définition dynamique des états, des états étant créés, fusionnant, se scindant et disparaissant au cours de l'activité de l'automate. Ce point a commencé à être exploré ici pour obtenir des mouvements plus précis du bras sans que le temps d'apprentissage s'accroisse beaucoup.

Au-delà de l'algorithme lui-même, l'architecture de l'artefact contraint les possibilités de cet algorithme ; il convient donc d'étudier l'algorithme ET l'architecture sous-jacente en même temps : seule une architecture suffisamment riche peut effectuer des comportements riches. De même, en interaction avec son environnement, un animal possède un répertoire comportemental d'autant plus riche que son milieu est riche ; il faut donc s'attendre à avoir le même genre de choses pour un artefact fondé sur TD.

Les expériences de façonnage ont bien montré que tout comportement physiquement possible pour MAABAC peut être appris. Cela met l'accent sur les programmes de renforcement qui sont utilisés ; ces programmes s'inspirent des travaux de psychologie sur l'analyse du comportement. Nous nous intéressons également à l'apprentissage qui pourrait être réalisé par exemple en guidant le geste de MAABAC pour apprendre à effectuer sa tâche plus rapidement : il s'agit alors d'apprentissage supervisé qui doit interagir avec l'apprentissage par renforcement. Un autre mode d'apprentissage possible est l'imitation auquel s'intéresse plusieurs chercheurs [1].

Une fois un automate MAABAC réalisé, différentes applications sont envisagées du moment où son comportement est relativement réaliste. Ainsi, nous envisageons de l'utiliser pour créer des séquences d'images de synthèse. Ce

type d'automate peut également être utilisé pour contrôler des agents dans des jeux vidéos ou des agents explorant le web à la recherche d'informations. Plus généralement, bien qu'encore relativement jeunes, les algorithmes de renforcement à base de différence temporelle ont été utilisés à ce jour pour résoudre différents problèmes, dont des problèmes de contrôle. Leur potentiel pour résoudre des problèmes difficiles dans l'avenir, après quelques développements supplémentaires, paraît relativement clair. Le développement récents de travaux concernant la résolution de problèmes en variables continues est à cet égard très significatif [54].

Enfin, n'oublions pas l'intérêt de ce travail pour les sciences du comportement animal, motivation initiale de ce travail. En participant à la mise en place de modèle formel du comportement animal, ce travail peut espérer participer à l'amélioration de la connaissance du vivant.

## References

- [1] P. Andry, S. Moga, Ph. Gaussier, A. Revel, and J. Nadel. Imitation: learning and communication. In *Proc. From Animals to Animat (SAB) 6*, pages 353–362. MIT Press, 2000.
- [2] A.G. Barto. Reinforcement learning and adaptive critic methods. In D.A. White and D.A. Sofge, editors, *Handbook of intelligent control: neural, fuzzy, and adaptive approach*, pages 469–491. Van Nostrand Reinhold, 1992.
- [3] A.G. Barto, R.S. Sutton, and C.W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5):834–846, 1983.
- [4] H. Bersini. Immune network and adaptive control. In F.J. Varela and P. Bourgine, editors, *Towards a practice of autonomous systems, Proc. of the First European Conf. on Artificial Life*, pages 217–226. MIT Press, 1991.
- [5] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.
- [6] E. Bonabeau, M. Dorigo, and G. Théraulaz. *Swarm Intelligence, from Natural to Artificial Systems*. Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press, 1999.
- [7] E. Bonabeau and G. Théraulaz, editors. *Intelligence Collective*. Hermes, 1994.
- [8] F. Bousquet, Ch. Cambier, and P. Morand. Distributed artificial intelligence and object-oriented modelling of a fishery. *Mathematical Computing Modelling*, 20(8):97–107, 1994.

- [9] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2:14–23, 1986.
- [10] R.A. Brooks. *Cambrian Intelligence: The early history of the new AI*. MIT Press, 1999.
- [11] C. Cambier. *Simdelta: un système multi-agents pour simuler la pêche sur le Delta Central du Niger*. PhD thesis, Université de Paris 6, July 1994.
- [12] C. Cassagnabère. Modélisation et simulation du comportement adaptatif d'un bras virtuel lors d'un mouvement d'atteinte, 2001. DEA d'informatique.
- [13] A.C. Catania. The taxonomy of verbal behavior. In Lattal and Perone, editors, *Handbook of research methods in human operant behavior*, pages 405–433. Plenum, 1998.
- [14] D.J. Chalmers. The evolution of learning: An experiment in genetic connectionism. In D.S. Touretzky, J.L. Elman, T.J. Sejnowski, and G.E. Hinton, editors, *Proc. of the 1990 Connectionist Models Summer School*. Morgan Kauffman, 1990.
- [15] C. Chang and P. Gaudiano. Application of biological learning theories to mobile robot avoidance and approach behaviors. *Journal of Complex Systems*, 1:79–114, 1998.
- [16] C. Clément. *L'ontogénèse du contrôle temporel du comportement humain envisagé comme un système dynamique non linéaire*. PhD thesis, Université de Lille 3, URECA, Villeneuve d'Ascq, October 1999. Thèse de doctorat de Psychologie.
- [17] P. Coquillard and D.R.C. Hill. *Modélisation et simulation d'écosystèmes — Des modèles déterministes aux simulations à événements discrets*. Masson, 1997.
- [18] D.L. DeAngelis and L.J. Gross. *Individual-based models and approaches in ecology*. Chapman & Hall, 1994.
- [19] S. Delepouille. *Coopération entre agents adaptatifs ; étude de la sélection des comportements sociaux, expérimentations et simulations*. PhD thesis, Université de Lille 3, URECA, Villeneuve d'Ascq, October 2000. Thèse de doctorat de Psychologie.
- [20] S. Delepouille, Ph. Preux, and J-C. Darcheville. L'apprentissage par renforcement comme résultat de la sélection. *Extraction des connaissances et apprentissage*, 1(3):9–30, 2001.
- [21] M. Dorigo and M. Colombetti. *Robot Shaping: An experiment in behavior engineering*. MIT Press, 1998.

- [22] R. Duboz, E. Ramat, and Ph. Preux. Towards a coupling of continuous and discrete formalisms in ecological modelling - influences of the choice of algorithms and results. In N. Giambiasi and C. Frydman, editors, *Proc. 13th European Simulation Symposium*, pages 481–487, October 2001.
- [23] G. Edelman. *Biologie de la conscience*. Odile Jacob, 1992.
- [24] G.M. Edelman. *Neural Darwinism*. Basic Books, 1987.
- [25] J. Ferber. *Les systèmes multi-agents*. Inter-Éditions, 1995.
- [26] D. Floreano. Emergence of nest-based foraging strategies in ecosystems of neural networks. In *Proc. SAB 2*, pages 410–416, 1993.
- [27] D. Floreano and F. Mondada. Evolution of plastic neurocontrollers for situated agents. In P. Maes, M. Mataric, J.A. Meyer, J. Pollack, H.L. Roitblatt, and S.W. Wilson, editors, *Proc. SAB 4*, pages 402–410. MIT Press, 1996.
- [28] D. Floreano and S. Nolfi. Learning and evolution. *Autonomous robots*, 7(1):89–113, 1999.
- [29] S. Forrest, B. Javornik, R.E. Smith, and A.S. Parelson. Using genetic algorithms to explore pattern recognition in the immune system. *Evolutionary Computation*, 1, 1993.
- [30] S. Forrest and T. Jones. Modeling complex adaptive systems with Echo: Mechanisms of adaptation. In R.J. Stonier and X.H. Yu, editors, *Complex Systems: Mechanisms of Adaptation*, pages 3–21. IOS Press, 1994.
- [31] P. Grassé. La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes natalensis* et *Cucitermes* sp. la théorie de la stigmergie : essai d'interprétation des termites constructeurs. *Insectes sociaux*, 6:41–83, 1959.
- [32] V. Grimm. Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological Modeling*, (115):129–148, 1999.
- [33] S. Grossberg. On the dynamics of operant conditioning. 33:225–255, 1971.
- [34] A. Guillot and J-A. Meyer. From SAB94 to SAB2000: What's new, animat? In *Proc. SAB 2000*, pages 3–12. MIT Press, 2000.
- [35] A. Guillot and J-A. Meyer. The animat contribution to cognitive systems research. *Journal of Cognitive Systems Research*, 2(2):157–165, 2001.
- [36] D.F. Hake and D.R. Olvera. Cooperation, competition, and related social phenomena. In A. C. Catania and T. A. Brigham, editors, *Handbook of applied behavior analysis*, pages 208–245. New-York: Irvington, 1978.

- [37] D. Hill, P. Coquillard, J. De Vaugelas, and A. Meinez. A stochastic model with spatial constraints: Simulation of *caulerpa taxifolia* development in north-mediterranean sea, 1995.
- [38] J.H. Holland. Outline of a logical theory of adaptative systems. *Journal of the ACM*, 7:297–316, November 1961.
- [39] J.H. Holland. *Adaptation in Natural and Artificial Systems*. Michigan Press University, Ann Arbor, MI, 1975.
- [40] J.H. Holland. *Adaptation in Natural and Artificial Systems*. A Bradford Book. MIT Press, second edition, 1992. ISBN: 0-262-58111-6.
- [41] J.H. Holland. *Hidden Order — How adaptation builds complexity*. Helix Books. Addison-Wesley Publishing Company, 1995.
- [42] J. Joséfowicz. ? PhD thesis, Université de Lille 3, URECA, Villeneuve d’Ascq, 2001. thèse de doctorat de psychologie, à soutenir en décembre 2001.
- [43] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [44] J. Kodjabachian. *Développement et évolution de réseaux de neurones artificiels*. PhD thesis, Paris 6, 1998.
- [45] P.N. Kugler and M.T. Tarvey. *Information, natural law, and the self-assembly of rythmic movements*. Hillsdale, 1987.
- [46] H. Lejeune, A. Ferrara, F. Simon, and J.H. Wearden. Adjusting to change in the time of reinforcement: peak interval transition in rats. *Journal of Experimental Psychology: Animal Behavior Processes*, 23:311–331, 1997.
- [47] M.L. Littman. Simulations combining evolution and learning. In *in [52]*, pages 465–477. 1996.
- [48] D. McFarland and T. Boesser. *Intelligent behavior in animals and robots*. MIT Press, 1993.
- [49] G. Metta, R.E.S Manzotti, F. Panerai, and G. Sandini. Development: is it the right way towards humanoid robotics? In *IAS-6*, July 2000.
- [50] J.A. Meyer and S. Wilson. From animals to animats. In *Proc. From Animals to Animat (SAB) 1*. MIT Press, 1991.
- [51] O. Miglino, S. Nolfi, and D. Parisi. Discontinuity in evolution: how different levels of organization imply pre-adaptation. In *in [52]*. 1996.
- [52] M. Mitchell and R. Belew, editors. *Adaptive Individuals In Evolving Population Models*. Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley Publishing Company, 1996.

- [53] Tom M. Mitchell. *Machine Learning*. Mc Graw-Hill, 1997.
- [54] R. Munos and A. Moore. Variable resolution discretization in optimal control. *Machine Learning*, 2001.
- [55] D. Parisi and S. Nolfi. The influence of learning on evolution. In *in [52]*, pages 419–428. 1996.
- [56] D. Parisi, S. Nolfi, and F. Cecconi. Learning, behavior, and evolution. In F.J. Varela and P. Bourguine, editors, *Towards a practice of autonomous systems, Proc. of the First European Conference on Artificial Life (ECAL)*, pages 207–216. MIT Press, 1991.
- [57] R. Pfeifer, B. Blumberg, J-A. Meyer, and S.W. Wilson, editors. *Proc. Fifth International Conference on Simulation of Adaptive Behavior (SAB 5)*. MIT Press, 1999.
- [58] R. Pfeiffer and C. Scheier. *Understanding intelligence*. MIT Press, 1999.
- [59] J. Randaløw and P. Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. 1998.
- [60] G.N. Reeke, O. Sporns, and G.M. Edelman. Synthetic neural modelling: Comparisons of population and connectionist approaches. In R. Pfeifer, Z. Schreter, F. Fogelman-Soulié, and L. Steels, editors, *Connectionism in Perspective*. Elsevier Science Publishers, 1989.
- [61] R.A. Rescorla and A.R. Wagner. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In *Classical conditioning*, volume 2: Current research and theory. Prentice-Hall, 1972.
- [62] C.W. Reynolds. Flocks, herds and schools: a distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.
- [63] L.M. Saksida, S.M. Raymond, and D.S. Touretzky. Shaping robot behavior using principles from instrumental conditioning. *Robotics and autonomous systems*, 22(3/4), 1998.
- [64] B. Scassellati. Building behaviors developmentally: a new formalism, 1999.
- [65] B.F. Skinner. *The behavior of organisms*. Appleton-Century Crofts, 1938.
- [66] B.F. Skinner. Selection by consequences. *Science*, 213:501–514, 1981.
- [67] E. Spier and D. McFarland. Learning to do without cognition. In *[57]*, pages 38–47, 1998.
- [68] J. Staddon. *The new behaviorism: Mind, Mechanism, and Society*. Psychology Press, 2001.

- [69] J.E.R. Staddon. *Adaptive behavior and learning*. Cambridge University Press, 1983.
- [70] R.S. Sutton. Learning to predict by the method of temporal difference. *Machine Learning*, 3:9–44, 1988.
- [71] R.S. Sutton and A.G. Barto. Time-derivative models of pavlovian reinforcement. In M. Gabriel and J. Moore, editors, *Learning and Computational Neurosciences: Foundations of Adaptive Networks*, pages 497–537. MITP, 1990.
- [72] R.S. Sutton and A.G. Barto. *Reinforcement learning: an introduction*. MIT Press, 1998.
- [73] E. Thelen and L.B. Smith. *A dynamic systems approach to the development of cognition and action*. MIT Press, 1994.
- [74] E.L. Thorndike. Animal intelligence: An experimental study of the associative process in animals. *Psychology Monographs*, 2, 1898.
- [75] E.L. Thorndike. *Animal Intelligence: Experimental Studies*. Mac Millan, 1911.
- [76] D.S. Touretzky and L.M. Saksida. Skinnerbots. In M. J. M. P. Maes, J.-A. Meyer, J. Pollack, and S. W. Wilson, editors, *From Animals to Animats 4: Proceedings of the fourth international conference on simulation of adaptive behavior*. Cambridge, MA: The MIT Press/Bradford Books, 1996.
- [77] F.J. Varela, E. Thompson, and E. Rosch. *L’inscription corporelle de l’esprit – Sciences cognitives et expérience humaine*. Seuil, coll. La couleur des idées, 1993.
- [78] G. Walter. An imitation of life. *Scientific American*, pages 42–45, May 1950.
- [79] C.J.C.H. Watkins. *Learning from delayed rewards*. PhD thesis, King’s college, Cambridge, UK, 1989.
- [80] M.W. Williamson. Postural primitives: interactive behavior for a humanoid robot arm. In P. Maes, M. Mataric, J.A. Meyer, J. Pollack, H.L. Roitblatt, and S.W. Wilson, editors, *Proc. SAB 4*, pages 124–131. MIT Press, 1996.