



HAL
open science

Apprentissage Incrémental de Règles Sensorimotrices dans un Robot, du Babillage Moteur à l'Utilisation d'Outils

Raphaël Braud

► **To cite this version:**

Raphaël Braud. Apprentissage Incrémental de Règles Sensorimotrices dans un Robot, du Babillage Moteur à l'Utilisation d'Outils. Robotique [cs.RO]. université de cergy-pontoise, 2017. Français. NNT: . tel-01724183

HAL Id: tel-01724183

<https://hal.science/tel-01724183>

Submitted on 13 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Université de Cergy-Pontoise - Ecole doctorale Sciences et Ingénierie

THÈSE DE DOCTORAT

présentée et soutenue publiquement pour obtenir le titre de docteur en Robotique et Sciences Cognitives

APPRENTISSAGE INCRÉMENTAL DE RÈGLES
SENSORIMOTRICES DANS UN ROBOT,
DU BABILLAGE MOTEUR À L'UTILISATION D'OUTILS

par

Raphaël Braud

ETIS - ENSEA / Université de Cergy-Pontoise / CNRS UMR 8051

Soutenue le 23 novembre 2017, devant le jury composé de :

O. SIGAUD,	Professeur des Universités, ISIR/UPMC	Rapporteur
A. REVEL,	Professeur des Universités, Université de La Rochelle	Rapporteur
J. K. O'REGAN,	Directeur de Recherche, LPP/CNRS	Examineur
F. ALEXANDRE,	Directeur de Recherche, INRIA de Bordeaux	Examineur
P. GAUSSIER,	Professeur des Universités, Université de Cergy-Pontoise	Directeur de thèse
A. PITTI,	Maître de Conférences, Université de Cergy-Pontoise	Encadrant de thèse

Cette thèse porte sur l'intégration développementale de différents systèmes d'apprentissage dans un robot, du babillage moteur à l'émergence de l'utilisation d'outils. L'utilisation d'outils recouvre de nombreuses problématiques, certaines bas niveau (comme l'extension du schéma corporel) et d'autres plus haut niveau (comme la capacité à faire une séquence d'actions).

Nous avons pour cela proposé un modèle appelé Dynamic Sensorimotor Model (DSM). DSM apprend des lois sensorimotrices, qui consistent à prédire les variations sensorielles (comme le déplacement d'un objet dans l'espace visuel) en fonction :

1. De magnitudes motrices (comme des commandes en vitesse de servomoteurs).
2. D'un contexte donné (un vecteur de données sensorielles).

Un tel prédicteur peut apprendre et affiner ses lois sensorimotrices dans n'importe quelle situation, que ce soit durant l'exécution d'une tâche ou durant une phase de babillage moteur. L'apprentissage de ces prédictions est donc indépendant de l'exécution de tâches particulières, et pourra être exploité dans de nouveaux contextes, et pour satisfaire de nouvelles tâches.

Pour cela, DSM contient un mécanisme de simulation motrice mais aussi un mécanisme de simulation de contextes. Ces simulations portent ainsi sur :

1. Les entrées motrices, ce qui permet de déterminer les commandes motrices à effectuer en vue d'une tâche particulière.
2. Les entrées sensorielles, ce qui permet de proposer des contextes alternatifs au sein desquels les actions permettant la réalisation d'une tâche pourront être effectuées. Ces contextes alternatifs pourront alors se constituer en sous-buts permettant d'effectuer une séquence d'actions.

Grâce à ces simulations, des expériences sur robot réel ont permis de satisfaire une tâche consistant à rejoindre une cible avec l'extrémité du bras, en faisant un détour pour saisir un outil. La saisie a comme propriété d'étendre le schéma corporel (le segment terminal du bras du robot).

La capacité à faire des séquences à la volée repose sur les contextes qui auront été appris. Cela met en évidence l'importance d'avoir des contextes ne contenant que les données suffisantes à la prédiction, afin de générer, par le mécanisme de simulation, des sous-buts les plus minimaux possibles pour satisfaire un but donné.

Notre modèle catégorise des lois additives afin de ne pas perturber les lois sensorimotrices précédemment apprises et ainsi apprendre des lois de manière incrémentale. Dans DSM, une nouvelle catégorie se caractérise par l'instauration d'une distance entre la configuration sensorielle correspondant au contexte actuel, dans lequel les lois courantes sont en échec, et le dernier contexte dans lequel ces lois s'appliquaient correctement. Cette distance entre contextes est donc multimodale, et indépendante de la topologie propre des senseurs d'entrée. Par contre, étant issue de deux situations à deux moments différents, cette distance dépend de l'exploration sensorimotrice du robot durant cet intervalle de temps. Pendant cette période, les senseurs qui auront suffisamment changés de valeurs apparaîtront comme discriminant un contexte par rapport à l'autre, bien qu'ils ne soient pas tous pertinents. Ce sera par l'action que les senseurs pertinents seront sélectionnés.

Inspired by concepts found in developmental psychology, my work focuses on robotic learning through motor babbling in order to achieve low-level sensorimotor control and, subsequently, to progress to more high-level behaviours such as the use of tools.

Tool-use raises several key issues related to the extension of the body schema and the ability to make sequences of actions. In this presentation I will discuss my research efforts in this area by presenting a model called "Dynamic Sensorimotor Model (DSM)". DSM learns sensorimotor laws by making predictions about sensory input variations, as a result of observing environmental phenomena and interacting with objects in the reaching space.

Sensorimotor laws depend on; 1) motor magnitudes (e.g., motor commands in velocity) and 2) a given context (i.e., a sensory input vector). A predictor learns and refines sensorimotor laws either during the execution of a task or during a motor babbling phase. Learning laws is therefore independent of the execution of specific tasks and they can be exploited in both new contexts and/or for new tasks.

DSM employs two mechanisms. First, a mechanism for motor simulations that considers the result of simulated motor inputs to determine appropriate motor commands to be performed towards a particular task. Second, a mechanism for context simulations that uses simulated sensory inputs in order to identify contexts that can potentially form sub-goals towards the completion of a task.

The performance of the system is evaluated through a series of experiments conducted using both a simulated and a real robotic platform. The results demonstrate the ability of the system to complete reaching tasks and highlight its strength in making use of a nearby tool when the target is not within its reach.

The ability to make sequences of actions on the fly is based on the accuracy of the contexts that the system gradually learns. The last part of my work focuses on improving the efficiency of making sequences of actions by offering the ability to categorize contexts based on the variations observed in the sensors with respect to the variation of the sensorimotor laws.

Table des matières

1	État de l’art	13
1.1	Approches psychologiques, cognitives et neurobiologiques de l’utilisation d’outils et de l’apprentissage sensorimoteur	14
1.1.1	Utilisation d’outils	14
1.1.2	Théorie des contingences sensorimotrices	17
1.1.3	Principe idéomoteur	20
1.1.4	Séquence d’actions	21
1.1.5	Affordances, neurones miroirs et théorie de la simulation	23
1.2	Apprentissage en robotique	29
1.2.1	Apprentissage sensorimoteur	29
1.2.2	Théorie du contrôle optimal	32
1.2.3	Apprentissage par renforcement	36
1.2.4	Approche PerAc	37
1.2.5	Utilisation d’outils en robotique	41
1.2.6	Apprentissage autonome et “open-ended”	44
1.3	Conclusion	46
2	Apprentissage de couplages sensorimoteurs	47
2.1	Modèle “absolu” couplant des positions dans l’espace visuel et moteur	47
2.1.1	Apprentissage visuo-moteur	48
2.1.2	Application à l’utilisation d’outils	50
2.1.3	Conclusion	53
2.2	Transformations sensorimotrices et neurones à champ de gains	55
2.2.1	Modèle	56
2.2.2	Apprentissage de transformations sensorimotrices	58
2.2.3	Expériences	59
2.2.4	Résultats	60
2.2.5	Conclusion	62
2.3	Modèle couplant flot optique et mouvements oscillatoires	65
2.3.1	Primitives visuelles et motrices	65
2.3.2	Apprentissage visio-moteur et imitation bas niveau	67
2.3.3	Apprentissage de la forme du bras pendant le babillage moteur	69
2.3.4	Ambiguïté visuelle et émergence d’un comportement de “reaching”	70

2.3.5	Conclusion	71
2.4	Conclusion	74
3	Modèle relatif, simulations et réalisation de séquences	75
3.1	Modèle relatif, et comparaison avec un modèle absolu	76
3.1.1	Modèle relatif prédisant des variations sensorielles	76
3.1.2	Modèle “absolu” utilisé pour la comparaison	81
3.1.3	Comparaison de l’encodage de l’information effectué par les différentes stratégies	82
3.1.4	Expériences d’atteintes avec des stratégies “relative” et “absolue”	86
3.1.5	Discussion et conclusion	92
3.2	Séquences d’actions, basées sur un modèle relatif et des simulations	96
3.2.1	Introduction	96
3.2.2	Le Simulateur de Lois Sensorimotrices (SLS)	97
3.2.3	Architecture du modèle sensorimoteur dynamique (DSM)	99
3.2.4	Expériences sur robot réel	103
3.2.5	Discussion et conclusion	109
4	Apprentissage de lois causales incrémentales	117
4.1	Analyse du problème	120
4.1.1	Propriétés du système étudié	120
4.1.2	Conséquences et problèmes de l’actuelle catégorisation	123
4.1.3	Proposition	125
4.2	Modèle pour la régression dynamique de lois sensorimotrices	128
4.2.1	Principe	128
4.2.2	Fonctionnement, et intégration à DSM	130
4.3	Expériences	140
4.3.1	Mécanisme de recrutement	141
4.3.2	Performances et limitations	152
4.4	Discussion et conclusion	161
4.4.1	Remarques générales	161
4.4.2	Densité du réseau, et création d’unités sensorielles et motrices	161
4.4.3	Processus maturationnel et exploration active	162
4.4.4	Limitations et améliorations possibles	163
	Conclusion	167

Introduction

Lors de l'émergence de l'intelligence artificielle (IA) dans les années 1950, la majeure partie des travaux adoptait une approche computationnaliste et symbolique. Les robots devaient, quant à eux, être dotés de 3 capacités : d'une part sentir, ou percevoir, les informations provenant des capteurs, d'autre part raisonner et planifier le ou les actions en vue d'un but, afin, en dernier lieu, de les exécuter par les activations motrices correspondantes. Les robots étaient ainsi conçus pour effectuer, de manière pré-programmée, des tâches répétitives dans des environnements maîtrisés. De tels robots sont efficaces pour accomplir des tâches précises, mais cette aptitude repose sur la connaissance pointue des contraintes du problème à résoudre, la maîtrise de l'environnement du robot pendant l'exécution de ses tâches, ainsi que sur l'expertise des ingénieurs programmant ces robots. Si les résultats de cette approche sont impressionnants (pensons aux victoires, symboliques, de machines contre des humains, aux échecs puis au go), elle n'est pas pour autant dénuée de désavantages.

Puisque cette approche repose sur la capacité à prédire les difficultés qui se poseront et sur le contrôle de l'environnement, des problèmes fondamentaux apparaissent lorsque l'on attend du robot une interaction avec l'environnement réel, qui en viennent à remettre en cause l'approche symbolique même. En effet cette conception de l'IA vient, en raisonnant par manipulation de symbole, se heurter à la difficulté qu'il y a à faire le pont entre symboles et monde réel, problème appelé "Symbol Grounding Problem" [Harnad, 1990]. Dans le cas où les symboles précèdent l'appréhension du réel, la question de leur adéquation *a posteriori* avec le réel, et donc de leur signification effective, pose problème.

Or la question d'un pont effectif avec l'environnement réel, complexe et changeant, ne peut faire l'objet d'une impasse, surtout dans le cadre des attentes actuelles de la robotique dans le domaine des interactions sociales, de la vie économique, ou même de notre quotidien. Il y a au contraire un besoin d'autonomisation qui passe par l'absence d'*a priori* forts sur l'environnement. Car de nombreux changements, potentiellement radicaux, de l'environnement ou même de la morphologie du robot (comme c'est le cas lors de l'utilisation d'outil) sont impossibles à prévoir de manière exhaustive lors du travail de spécification du problème à résoudre qui est effectué en amont par les ingénieurs. Soulignons que ce travail deviendrait d'ailleurs de plus en plus infaisable si en plus chaque problème devait faire l'objet d'une solution particulière. Peu à peu se dessine donc l'idéal d'un robot à même de satisfaire des tâches non préprogrammées à l'avance, par un apprentissage et des adaptations tout au long de son existence.

Le besoin d'envisager de nouvelles approches se trouve également illustré, et expliqué par l'aspect contre-intuitif du paradoxe de Moravec [Moravec, 1988], lequel affirme qu'il est, comparativement, plus facile de reproduire à l'aide d'ordinateurs des raisonnements du niveau de

ceux d'un adulte, pour ce qui concerne les tests d'intelligence ou le jeu d'échec, que de reproduire les compétences d'un enfant d'un an, quand il s'agit de perception ou de mobilité.

De fait, une nouvelle IA a émergé (voir notamment [Brooks, 1991]), mettant le corps et l'interaction sensorimotrice au cœur de son approche. Celle-ci est inspirée par les êtres vivants, et leurs comportements adaptatifs en interaction avec le monde extérieur. La théorie de l'énaction [Francisco et al., 1991] place ainsi au centre cette interaction, et se constitue de fait comme une approche alternative au computationnalisme. Cette approche est basée sur la théorie des systèmes autopoïétiques [Varela et al., 1974], lesquels se définissent par une capacité à se redéfinir eux-mêmes matériellement, en permanence, de part un jeu d'interaction (intégration et différenciation) avec l'environnement. Cet aspect essentiel du corps dans l'IA se retrouve dans le courant de la cognition incarnée ("embodied cognition"), lequel donne toute son importance au rôle du corps, à l'expérience et au vécu de celui-ci, pour comprendre les mécanismes du cerveau.

Quitter la vision dualiste pour aborder la dynamique d'un système se définissant en permanence par son interaction avec ce qui l'entoure pose de nombreuses questions, lesquelles font notamment l'objet de l'approche sensorimotrice. Cette approche amène à questionner les concepts fondamentaux de la robotique, notamment en gommant, ou tout du moins en jetant le flou sur la frontière pourtant claire, dans l'esprit d'un roboticien, entre senseurs, et moteurs. Citons notamment la théorie des contingences sensorimotrices [O'Regan and Noë, 2001], qui place l'action au cœur du processus de perception (et sur laquelle nous reviendrons en détail dans notre état de l'art).

L'histoire de l'IA, telle que nous l'avons brièvement esquissée, reflète un tournant épistémologique en ce qu'une part non négligeable de sa communauté fonde de plus en plus sa recherche sur des questions fondamentalement pluridisciplinaires (touchant notamment la psychologie, les sciences cognitives, la neurobiologie, etc.). Notons qu'une telle ouverture sur d'autres sciences ne va bien entendu pas sans problèmes méthodologiques quant aux objectifs précis de cette science, à l'identification de ses questions fondamentales et à la mise en place de critères pertinents d'évaluation.

Le principe qui nous guidera sera de chercher dans quelle mesure les sciences du vivant, et la robotique, peuvent mutuellement s'enrichir. Pour ce faire, notre approche visera à proposer des architectures génériques résolvant des classes de problèmes variées plutôt que de proposer un ensemble de systèmes isolés résolvant indépendamment des problèmes donnés. Un des espoirs sous-jacent à cette approche est qu'à terme, intérêt scientifique (compréhension du vivant ou de l'humain) et efficacité pratique se rejoignent.

Dans cette visée, le but de cette thèse consiste à aborder la problématique de l'utilisation d'outils en suivant une approche sensorimotrice. On cherchera des propriétés d'adaptation à de nouvelles lois, à de nouveaux moteurs et senseurs, tout en faisant évoluer ces lois en fonction de contextes, et en étant à même de mobiliser ces contextes pour satisfaire des tâches impliquant l'utilisation d'outils. L'utilisation d'outils semble être un mécanisme clé, et se trouve de fait à la croisée de nombreuses problématiques et disciplines, notamment en psychologie, en neurosciences, en robotique, mais aussi en philosophie, en anthropologie ou encore dans les sciences du langage (voir Sec. 1.1.1). Cette capacité paraît centrale à une forme d'intelligence, et révélatrice d'une propriété que l'on soupçonne fondamentale. Cette question a notamment été reliée à la définition même de l'homme, ou encore à celle de l'émergence du langage (lui aussi souvent invoqué comme propriété fondamentale de l'être humain), bien que paradoxalement cette capacité se retrouve également chez les animaux. Un sous-ensemble de ces problématiques sera

développé dans l'état de l'art, mais notre travail lui se concentrera sur la modélisation de mécanismes, implémentables sur robot, susceptibles de donner lieu à l'émergence d'un comportement de type utilisation d'outils.

Dans la continuité de ce que nous avons dit, l'efficacité pratique dont il sera question et qui sera poursuivie dans cette thèse sera plus celle de la recherche d'un codage sensorimoteur tentant d'être économe dans sa capacité à répondre à de multiples attentes (notamment celle de l'utilisation d'outils), que l'efficacité au sens de système performant mais dans un sous-ensemble restreint de ces attentes.

L'apprentissage sensorimoteur, et plus précisément l'encodage de l'information sensorimotrice sera donc notre problématique majeure. On s'interrogera sur ce qui est appris, sur la capacité de cet apprentissage à se développer et s'enrichir au cours du temps, ainsi que sur les effets de la boucle sensorimotrice elle-même sur ce qui est appris : dans la mesure où cette boucle est à l'origine de perceptions, on s'interrogera sur l'influence de cette boucle sur la perception de ce qu'est un senseur, un moteur, ou la loi elle-même.

L'aspect dynamique de la boucle sensorimotrice nous amènera rapidement à des questionnements sur ce qui est dynamique, et qui forme les lois sensorimotrices à apprendre. Nous nous interrogerons donc sur ce qui varie, ou co-varie : les moteurs, les senseurs, ainsi que les lois de co-variations elles-mêmes, et leurs contextes d'application.

Dans le premier chapitre (Sec. 1), nous présenterons un état de l'art des différentes théories et modèles pertinents pour notre sujet, aussi bien en psychologie et en sciences cognitives qu'en robotique et en intelligence artificielle, ce qui permettra de présenter différents travaux qui ont inspiré cette thèse. Ce chapitre introduira notamment la différenciation entre ce qu'on appelle stratégie absolue, et stratégie relative, pour l'encodage sensorimoteur. Cette caractérisation en 2 grandes catégories d'encodages (que l'on peut résumer dans l'apprentissage de coordonnées fixes, ou de co-variations) servira de base de réflexion dans la suite de la thèse.

Dans le second chapitre (Sec. 2), nous expérimenterons plusieurs stratégies d'encodages et observerons les avantages et inconvénients de chacune, notamment du point de vue de l'utilisation d'outils. Nous commencerons par un apprentissage qui suit une stratégie absolue, avant d'examiner deux autres modèles glissant progressivement vers des stratégies dites plus "relatives".

Dans le chapitre suivant (Sec. 3), nous proposerons un nouveau modèle suivant une stratégie relative, et comparerons ses caractéristiques et performances, pour une tâche d'atteinte, avec un modèle adoptant une stratégie absolue (et procédant par l'apprentissage de fonctions de bases radiales). Nous verrons par la suite comment les propriétés de l'encodage de ce modèle permettent d'effectuer des séquences d'actions non précédemment apprises ou parcourues, et impliquant l'utilisation d'outils pour atteindre une cible.

Dans le dernier chapitre (Sec. 4), nous nous interrogerons sur la capacité du modèle à apprendre de manière incrémentale des lois sensorimotrices au cours du temps, tout en limitant l'oubli de ce qui a déjà été appris. Nous enrichirons donc le modèle en ce sens, ce qui lui permettra notamment de déterminer, par l'action, les entrées sensorielles pertinentes ainsi que des lois d'activations propres, et relatives aux lois et interactions sensorimotrices passées.

État de l'art

Dans cette état de l'art, le but sera d'explorer différentes théories tant dans le domaine de la robotique que de la psychologie ou la neurobiologie. Notre problématique s'étend du babillage moteur jusqu'à l'utilisation d'outils : il s'agit plus d'étudier les mécanismes d'encodage de l'information sensorimotrice permettant d'aboutir *in fine* à une utilisation d'outils, que des mécanismes dédiés à cette seule fin. Nous étudierons donc, dans ces différents domaines, des approches importantes ainsi que certaines d'entre elles qui nous auront plus particulièrement inspirées dans nos travaux, et qui permettent d'appréhender ces questions avec un assez large spectre d'outils théoriques d'horizons différents.

Nous décomposerons ce chapitre en deux principales sections : la première fera un état de l'art relatif à notre problématique sous l'angle de la psychologie et de la neurobiologie, la seconde sous l'angle de l'apprentissage en robotique.

Dans chaque partie, nous commencerons par introduire nos problématiques principales, à savoir celle de l'encodage sensorimoteur en robotique (Sec. 1.2.1), et celle de l'utilisation d'outils telle que peuvent l'étudier les psychologues ou les neurobiologistes (Sec. 1.1.1).

Pour ces différentes disciplines nous aborderons ensuite les théories permettant de proposer des solutions à ces problématiques. Il s'agira des théories relatives au contrôle optimal, à l'apprentissage par renforcement ou encore à l'approche PerAc pour la question du point de vue robotique (Sec. 1.2.2-1.2.4). En ce qui concerne la psychologie, il s'agira des questions fondamentales relatives au lien entre senseur et moteur, ou bien encore à la capacité à planifier une séquence d'action (Sec. 1.1.2-1.1.4).

Pour finir, nous avons sélectionné un ensemble de théories ou mécanismes clés permettant d'embrasser pleinement la problématique de l'encodage sensorimoteur susceptible de permettre à un robot d'utiliser des outils. Nous aborderons des théories transversales comme celles des affordances, des neurones miroirs, ainsi que de la théorie de la simulation (Sec. 1.1.5). D'autre part, nous traiterons plus spécifiquement la question de l'utilisation d'outils en robotique (Sec. 1.2.5), puis la question plus générale de l'apprentissage autonome et "open-ended" en robotique (Sec. 1.2.6).

1.1 Approches psychologiques, cognitives et neurobiologiques de l'utilisation d'outils et de l'apprentissage sensorimoteur

Dans cette partie, nous avons sélectionné quelques éléments de la littérature de ces différents domaines proposant soit des résultats fondamentaux, soit des théories portant sur des mécanismes qui nous ont semblé pertinents pour les travaux de cette thèse. Il ne s'agira bien souvent que d'un bref aperçu ne prétendant nullement à l'exhaustivité, mais permettant de donner des éléments clés pour la modélisation.

Afin de délimiter au mieux le champs de notre recherche, nous commencerons par nous intéresser spécifiquement à la question de l'utilisation d'outils (Sec. 1.1.1) en tentant de donner un bref aperçu des problématiques dites bas niveaux et hauts niveaux auxquelles la littérature rattache cette question. Ce bref panorama nous permettra de parcourir ces problématiques dans les sections qui suivent.

Puisque la question de l'encodage sensorimoteur est première et constitue l'axe choisi pour aborder cette large question, nous extrairons ensuite un ensemble de points qui nous ont paru clés dans la théorie des contingences sensorimotrices (Sec. 1.1.2). Dans la continuité de cette question fondamentale du lien entre senseur et moteur, nous donnerons également un bref aperçu du principe idéomoteur (Sec. 1.1.3).

Afin d'embrasser plus largement la problématique de l'utilisation d'outil, suite à ces considérations fondamentales nous observerons rapidement les questions soulevées par la capacité à effectuer des séquences d'actions (Sec. 1.1.4), l'une des caractéristiques de l'utilisation d'outil.

Enfin nous examinerons diverses théories psychologiques basées sur des observations neurobiologiques, non spécifiques à l'utilisation d'outil bien qu'on les y rattache parfois, dont la transversalité à de nombreuses thématiques est riche en enseignements concernant certains mécanismes clés du cerveau. Nous traiterons ainsi des affordances, des neurones miroirs, et de la théorie de la simulation (Sec. 1.1.5).

1.1.1 Utilisation d'outils

La question de l'utilisation d'outils est à la croisée de différentes problématiques. C'est une question qui met en jeu une multitude de domaines, et les problématiques dans lesquelles elle nous plonge reflètent la nécessité d'une approche pluridisciplinaire pour apporter une réponse à la mesure du problème. La question de l'aptitude à utiliser des outils fait actuellement l'objet de recherches non seulement en robotique, pour permettre aux robots de satisfaire de nouvelles tâches, mais aussi en psychologie développementale pour comprendre les étapes nécessaires à la maîtrise de cette faculté et les mécanismes sous-jacents qui sont impliqués. De nombreuses études portent également sur les aires du cerveau mises en jeu dans l'obtention de cette faculté, et enfin, surplombant l'ensemble de ces aspects, la question philosophique du rapport entre l'utilisation d'outils et ce qui définit l'homme en tant qu'homme, question qui passe notamment par la question du langage, vient donner toute sa profondeur et sa portée à la problématique qui nous occupe.

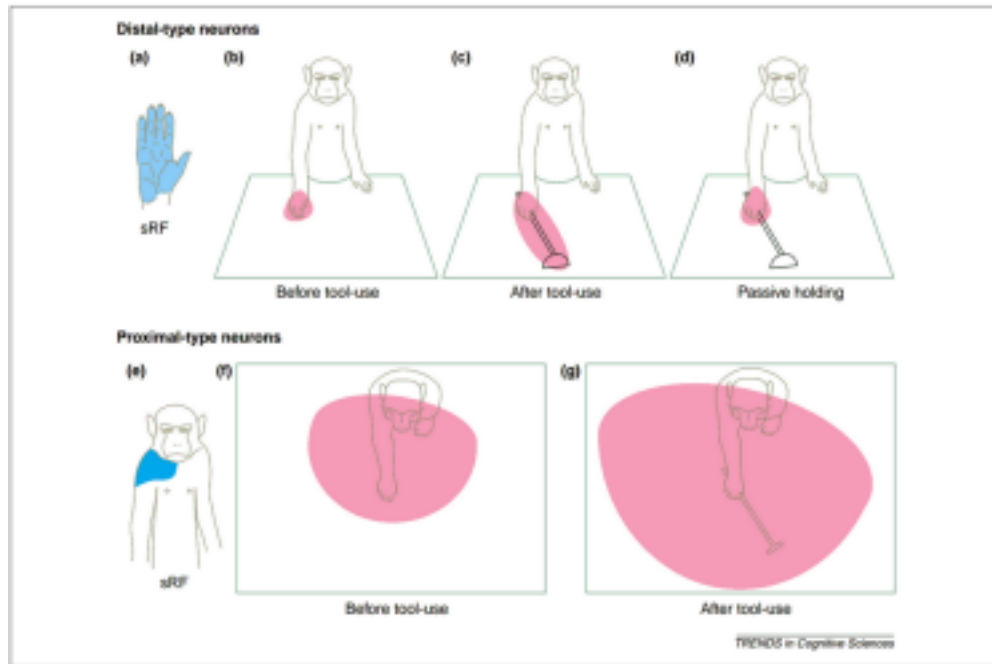


FIGURE 1.1 – Figure illustrant l’effet de l’utilisation d’outils sur le schéma corporel (position de la main, espace atteignable). Issu de [Maravita and Iriki, 2004]

Si, bien entendu, nous n’avons pas prétention à couvrir l’ensemble de ces domaines, et encore moins à y proposer une solution viable, c’est en gardant l’ensemble de ces questions à l’esprit que la réflexion a tenté d’être portée, afin d’en respecter l’étendue.

Dans cette section nous nous en tiendrons cependant à un résumé de la problématique d’un point de vue essentiellement psychologique.

Dans ce que l’on pourrait classer parmi les problématiques dites bas niveaux, l’utilisation d’outils suppose une extension du schéma corporel : des résultats montrent en effet, chez les macaques, que des neurones bimodaux répondant à la vision et à la proprioception réagissent différemment après l’utilisation active d’outils. En effet, des neurones codant la position de la main ont une activité étendue à la position de l’outil, et l’espace atteignable se trouve lui aussi étendu (voir figure 1.1). Ces résultats semblent indiquer qu’à très bas niveau l’outil est susceptible d’être perçu comme une extension du corps, et encode donc une extension du schéma corporel, différente selon les outils utilisés (voir les travaux de [Maravita and Iriki, 2004; Arbib et al., 2009] et de [Hoffmann et al., 2010] pour une analyse critique de la question et de la manière dont un tel schéma corporel peut être encodé dans un robot). Notons que la question reste ouverte de savoir dans quelle mesure un modèle interne de l’outil est appris, et dans quelle mesure c’est le modèle du bras lui-même qui est adapté [Kluzik et al., 2008].

L’aspect “bas niveau” est ici à comprendre comme étant fondamental, au sens où la question d’une telle plasticité dans l’apprentissage sera le fondement de toutes constructions ultérieures, que l’on pourrait alors qualifier de plus “haut niveau”, quand en définitif celles-ci dépendront principalement des caractéristiques de l’encodage sensorimoteur choisi. Or la question de ce qui doit être appris rejoint la question plus générale du lien entre senseurs et moteurs : il s’agit

d'une question non tranchée, et flirtant avec la philosophie en ce que prendre position dessus définit globalement une approche, parmi d'autres, au sein de différents courants de pensées faisant toujours l'objet de débats (voir par exemple [Jacob and Jeannerod, 2005; O'Regan and Block, 2012]). La question du lien entre senseurs et moteurs sera abordée dans les deux sections suivantes, en ayant donc le partie pris de ne nous intéresser dans cet état de l'art qu'à la théorie des contingences sensorimotrices, et au principe idéomoteur.

Parmi les caractéristiques qualifiées de "hauts niveaux", l'utilisation d'outils est aussi liée à des capacités cognitives de plus hauts niveaux (voir [Guerin et al., 2013]), et à des capacités d'abstraction telles que celles permettant de faire des séquences ([McCarty et al., 1999; Johnson, 2000; Johnson-Frey, 2004]), et est fortement liée aux problèmes du langage ([Stout and Chaminade, 2009; Cangelosi et al., 2010; Arbib, 2011; Steele et al., 2012]).

Dans les expériences menées par Fagard *et al.*, des enfants doivent utiliser un outil en forme de râtelier afin d'atteindre des jouets hors de portée. Les résultats indiquent que ce n'est seulement qu'entre le 16^{ème} et le 20^{ème} mois que les enfants commencent à intentionnellement essayer de rapprocher l'objet avec l'outil, ce qui suggère qu'une véritable compréhension de l'utilisation d'outils n'est pas acquise avant cet âge [Rat-Fischer et al., 2012]. Notons de plus que les performances de l'enfant se trouvent améliorées lorsque celui-ci comprend les intentions du démonstrateur lors d'un apprentissage par observation de cette tâche [Esseily et al., 2013].

Parmi les capacités notables que l'enfant développe avant ses 18 mois, on peut noter le comportement de moyen en vue d'une fin ("means-end") qui apparaît autour du 8^{ème} au 12^{ème} mois. Celui-ci implique l'exécution planifiée d'une séquence d'étapes permettant d'accomplir un but, et apparaît dans des situations dans lesquelles un obstacle doit être levé pour satisfaire le but [Willatts, 1999]. Ce comportement constitue une étape très importante et nécessaire pour l'utilisation d'outil, mais il est intéressant de noter que la capacité à planifier, et à faire des séquences, n'est pas suffisante pour l'utilisation d'outil que l'on observe plus tard, malgré l'aspect haut niveau des capacités qui sont pourtant requises.

Ainsi, même si la planification et l'exécution d'une séquence d'étapes permettant d'atteindre un but est un comportement complexe et fortement lié à l'utilisation d'outil, de tels comportements (requis pour ceux de "means-end") se développent avant l'apparition de celui d'utilisation d'outil, et ne sont donc pas suffisants. Notons également que l'apraxie idéatoire montre qu'une décorrélation est possible entre la capacité à effectuer certains mouvements, et la capacité à organiser ces mouvements en séquences visant un but précis, et ceci impacte la capacité à utiliser des outils [Osiurak et al., 2010]. La question plus spécifique de la capacité à faire des séquences sera ainsi abordée dans la section 1.1.4.

D'autres types de travaux ([Stout and Chaminade, 2007]), basés sur l'imagerie cérébrale, ont eux montré que, sur des sujets apprenant à faire des outils en pierre, la conceptualisation abstraite ou la planification stratégique n'est pas ce qui est le plus important quand il s'agit de l'utilisation d'outil, mais semble indiquer qu'il s'agit plutôt d'adaptation sensorimotrice, et de perception d'affordances. Cette idée est également développée dans [Osiurak et al., 2010], et nous explorerons des théories relatives aux affordances en section 1.1.5.

Pour finir nous noterons que, dans le cadre de cette thèse, la problématique de l'utilisation d'outils nous semble une bonne illustration de la problématique de l'"open-ended" telle qu'on la retrouve en robotique (et que nous développerons en section 1.2.6). En effet, un des cas qui semble typiquement refléter une capacité à utiliser un outil est lorsque celui-ci est entièrement nouveau (quelles que soient les différentes modalités qu'il sollicite) pour le robot : car autrement

il peut suffire, pour un roboticien, de l'encoder "en dur". Or si ça peut être le cas pour le bras ou la main, qui peuvent être interprétés comme les premiers outils dont nous disposons, nous pensons plutôt que c'est *d'une même manière* que l'usage d'un quelconque outil fait l'objet d'un apprentissage. Ainsi un robot aura à découvrir sans cesse de nouvelles aptitudes impactant ce qu'il est à même de faire, et donc, potentiellement, ayant également un impact sur ses précédents apprentissages sensorimoteurs. Il devra également faire appel à cette nouvelle aptitude en fonction du contexte, pour résoudre des tâches connues ou inconnues à l'avance. Notons que la capacité à mobiliser des connaissances liées à l'utilisation d'outil, au cours d'une tâche où l'outil n'est pas explicitement requis, sera notamment l'objet de cette thèse.

1.1.2 Théorie des contingences sensorimotrices

L'approche sensorimotrice se caractérise par un questionnement radical sur ce qu'est une perception. Elle cherche à caractériser précisément ce qui est en jeu lorsque, par exemple, nous percevons la couleur rouge, ou encore lorsque nous pressons une éponge. Elle s'intéresse donc à la question des qualia, c'est-à-dire la manière dont les choses nous apparaissent, les propriétés subjectives de l'expérience accessibles à la conscience mais non communicable autrement que par analogie avec autrui (ceux-ci diffèrent donc des représentations, ou des caractéristiques fonctionnelles).

Pour J.K. O'Regan et A. Noë [O'Regan and Noë, 2001], les explications purement neuronales (comme la conductance, la fréquence d'excitation ou même le réseau auquel appartient un neurone) ne peuvent rendre compte de l'expérience de ces qualia, au sens où cela donne lieu à un saut explicatif ("explanatory gap") entre ce qu'on appelle les corrélats neuronaux et l'expérience hautement subjective des qualia. Ils rejettent donc l'idée que l'explication des qualia réside dans ces corrélats neuronaux, ou encore que ceux-ci puissent tenir lieu de représentations internes agissant comme des symboles représentant nos perceptions.

Selon ces auteurs, l'expérience des qualia ne peut être réduite à un état, car elle provient d'une activité, ou encore d'une manière d'agir. Le rôle du cerveau n'est pas de générer une perception, mais de permettre l'*interaction* à laquelle nous donnerons le nom de perception.

Ainsi percevoir n'est pas un état, mais une activité. Celle-ci est basée sur la connaissance des contingences sensorimotrices, à savoir les liens observés entre les sensations et les activités motrices. La perception de la mollesse sera par exemple basée sur l'activité d'écraser une éponge (voir fig. 1.2). Ces liens sont appris par l'expérience : le cerveau analyse les conséquences des mouvements qu'il commande sur les sensations qu'il reçoit en retour. La boucle complète entre commandes motrices (générées par le cerveau), monde physique, et sensations (recueillies en dernier lieu par le cerveau) est requise dans cet apprentissage. Une fois les lois des contingences sensorimotrices maîtrisées, un individu est à même de percevoir car il est à même de s'appuyer sur cette connaissance pour explorer le monde : il ne s'agit plus alors d'une succession de processus passifs, mais de la capacité à modifier activement ses sensations. Pour le dire autrement, la passivité intrinsèque des sensations (de part leur nature de capteurs et non d'actionneurs) est compensée par la connaissance des lois régissant leurs variations, lesquelles sont alors vécues comme perceptions.

Afin de mieux comprendre ce qui est en jeu, nous nous proposons de différencier ici quatre niveaux d'abstractions (allant du plus bas au plus haut niveau) en détaillant ce qui nous semble

être les éléments clés de la théorie des contingences sensorimotrices, dans le cadre de cette thèse : newpage

- 1 Le premier niveau est celui des informations sensorielles réelles (sur lesquelles on ne peut tricher), à savoir l'impact de la réalité physique sur les senseurs (comme la stimulation rétinienne par exemple). Nous pouvons supposer que ce niveau est entièrement passif : il s'agit de sensations passives, à l'opposé de la perception qui sera, elle, active. (Notons qu'ici nous qualifierons de "réel" ce qui se rapporte à la réalité physique objective.).
- 2 Le second niveau est celui des données qui seront traitées pas les lois. Notons que ce niveau n'est pas explicitement décrit en tant que tel par les auteurs, mais nous supposons son existence de leur explication aux phénomènes des rêves, ou des hallucinations. Dans [O'Regan and Block, 2012], O'Regan propose deux explications à ces phénomènes : l'une des deux (notons que l'autre sera expliquée au niveau 4) est que le cerveau se retrouve parfois comme piégé, et ce pour différentes raisons, dans un état similaire à celui dans lequel il peut être lorsqu'une interaction réelle a lieu. Ceci a pour effet d'entraîner l'activation de lois non reliées au contexte réel courant. De fait, cet "état", en activant l'une ou l'autre loi sensorimotrice, peut ne pas découler du seul premier niveau. C'est pourquoi nous proposons l'existence, d'un point de vue fonctionnel (peu importe ici sa nature réelle), d'un niveau intermédiaire, susceptible d'être alors le lieu de la "tricherie" opérée sur le cerveau durant une hallucination ou un rêve.
- 3 Le troisième niveau, le plus important, est celui de l'apprentissage des lois régissant les contingences sensorimotrices. Le phénomène de perception est caractérisé par la loi sensorimotrice qui s'applique (que celle-ci soit réelle ou pas, comme dans le cas du rêve). Les quatre points qui suivent décrivent plus spécifiquement les propriétés de ce niveau.



FIGURE 1.2 – Notre perception de la mollesse ne provient pas de l'excitation d'un ensemble de neurones, mais des propriétés régissant les lois sensorimotrices qui s'appliquent lors de l'interaction entre la main qui presse et l'éponge (sensation des doigts qui s'enfoncent, etc.). Schéma issu de [O'Regan, 2011].

Apprentissage de la structure du changement : Une contingence sensorimotrice est une dépendance, observée durant l'interaction avec le monde, entre des senseurs et des moteurs. L'apprentissage des lois régissant ces contingences sensorimotrices consiste en l'apprentissage de la structure du changement qui est observé lors de l'activation motrice. En appelant S l'ensemble des toutes les entrées sensorielles, M l'ensemble de toutes les commandes motrices, et E l'ensemble de tous les états possibles de l'environnement, Philipona *et al.* [Philipona *et al.*, 2003] en déduisent la relation $S = \psi(M, E)$ entre moteurs M , senseurs S et états E du monde. Les auteurs proposent de ne pas avoir un apprentissage du type $M = \phi(S)$ dont le but est le contrôle moteur, mais au contraire d'avoir un apprentissage dont le but est d'apprendre les conséquences sensorielles des commandes motrices, en s'intéressant à l'espace tangent $\{dS\}$ en un point $S_0 = \psi(M_0, E_0)$. Remarquant que la robustesse dépendra du fait que ψ soit localement linéaire, les auteurs identifient deux sous-espaces distincts : le premier est le sous-espace $\{dS\}_{dE=0}$ des variations sensorielles due aux seules commandes motrices, et le sous-espace $\{dS\}_{dM=0}$ des variations sensorielles due aux seuls changements de l'environnement. Ils remarquent de plus que $\{dS\} = \{dS\}_{dM=0} + \{dS\}_{dE=0}$.

Principe de compensabilité : Le principe à l'œuvre dans ces apprentissages est celui de compensabilité. Une compensation apparaît lorsque la variation sensorielle provenant de dM et celle provenant de dE s'annule. Les variations dM et dS sont alors perçues comme compensables. Cette propriété, qui s'apprend par l'expérience sensorimotrice, mène à percevoir le monde extérieur à travers ses propres mouvements : suivant les intuitions de H. Poincaré, l'idée est justement que cette apprentissage permettent de découvrir et de caractériser le monde extérieur. Ainsi, de nombreux travaux ont porté sur la manière d'inférer la dimensionnalité du monde extérieur, en se basant sur la dimensionnalité des variations sensorielles compensables (voir [Philipona *et al.*, 2003; Laflaquière, 2013]).

Interaction réelle : Comme nous l'avons déjà souligné, cet apprentissage a besoin de l'interaction réelle avec le monde. Cette interaction est saisie par les lois de la présence sensorielle réelle : "bodiliness" (capacité à changer les données sensorielles à l'aide de son corps), "insubordinateness" (les entrées sensorielles peuvent également varier sans l'aide du corps), "grabbiness" (les sensations, par des transitions soudaines par exemple, peuvent attirer l'attention). Ces lois peuvent être apprises, et constituent, en quelque sorte, les "marqueurs" du réel.

Le contexte détermine l'applicabilité : Le point précédent vaut pour l'apprentissage, toutefois notre cerveau est à même de détecter l'applicabilité d'une loi déjà apprise indépendamment de son application réelle courante (c'est-à-dire à travers la boucle d'interaction sensorimotrice, et donc l'action réelle). Par exemple, lorsque la main est prête à presser une éponge et que la loi relative à la pression de cette éponge a été apprise, son applicabilité peut être détectée [O'Regan and Noë, 2001].

- 4 Le quatrième niveau d'abstraction consiste en l'accès aux lois sensorimotrices ("cognizing", voir [O'Regan and Block, 2012]). Cet accès cognitif détermine si une personne est consciente d'une expérience. Dans le point précédent nous avons vu que le phénomène de perception est caractérisé par la loi sensorimotrice qui s'applique, que l'interaction soit

réelle ou pas (comme dans le cas du rêve). Nous ajoutons ici que la conscience de cette perception est donnée par l'accès cognitif à cette perception, et donc aux lois et catégories qui s'y rapportent. Notons qu'il y a deux sauts, ou écarts, possible entre la conscience et les lois réelles (physiques) qui s'appliquent. Le premier provient du fait que la conscience est *un certain* accès aux lois qui s'appliquent : cet accès fournit un ensemble de lois potentiellement différent de celui qui se trouvent être effectivement appliqué (et notamment, cet accès peut être "trompé", ce qui constitue la seconde manière d'expliquer le phénomène des rêves [O'Regan and Block, 2012], la première étant citée au niveau 2). Le second écart possible provient du fait que les lois effectivement appliquées sont elle-mêmes potentiellement différentes de celles qui devraient l'être au vu de l'interaction réelle avec l'environnement, de part l'existence du niveau 2. Toutefois, il faut souligner que si les lois accédées par la conscience peuvent être celles de la pression d'une éponge (perception de la mollesse), elles peuvent aussi être celles relatives aux marqueurs du réel ("grabiness" par exemple). De la sorte, si la conscience d'une perception est indépendante de la réalité de cette perception (pourtant à la source des qualia qui y sont associés), celle-ci peut avoir accès au fait que cette perception n'est pas réelle.

En conséquence de cette définition, nous proposons de qualifier d'activité consciente une activité où l'on accède aux lois de cette activité, et ainsi au fait d'*éprouver* un ensemble de lois sensorimotrices identifiées.

De manière parallèle à ces questionnements, il a également été développé des théories concernant un codage commun des senseurs et des moteurs, en observant la relation entre ceux-ci moins du point de vue des lois qui régissent leurs relations ou des perceptions qu'elles entraînent, qu'en considérant comment senseurs et moteurs se lient dans la perception d'un but.

1.1.3 Principe idéomoteur

Le principe idéomoteur [James, 1890] postule que les buts sont perçus comme la représentation anticipée des conséquences de l'action, et que les actions volontaires (et la planification de l'action) sont contrôlées par la représentation anticipée des effets désirés. L'idée d'encoder l'action

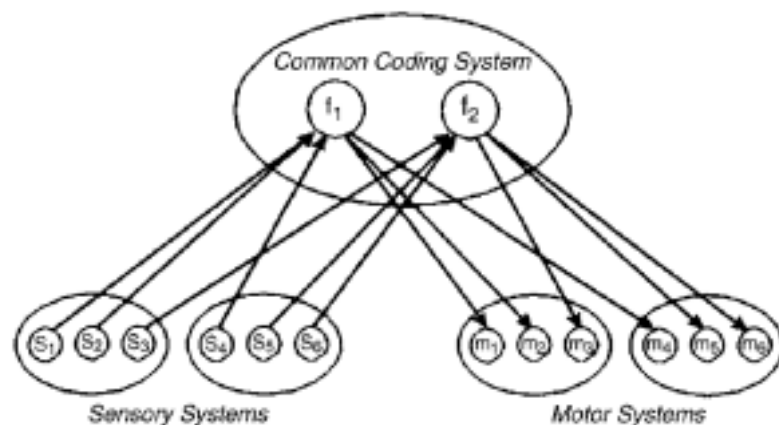


FIGURE 1.3 – Illustration de la théorie d'un codage commun entre plusieurs modalités de senseurs et de moteurs. Issu de [Hommel et al., 2001].

en terme de buts a notamment été développée avec la théorie du codage commun (“common coding”), qui propose que les actions soient codées en terme d’effets perceptifs produits [Prinz, 1990]. Dans la théorie de l’“event coding” (TEC) [Hommel et al., 2001], il est précisé que percevoir le stimulus d’un objet, et planifier une action volontaire, ne sont pas des processus distincts opérant sur des encodages différents mais sont fonctionnellement équivalents. Car ils ont tous les deux à faire à une représentation interne des événements externes. Dans la figure 1.3 cette représentation interne est illustrée, et on y voit que des stimuli provenant de différentes modalités (visuelle et auditive dans l’exemple des auteurs) sont codés ensembles, et utilisés pour activer à leur tour différents moteurs (comme la main ou la voix). Ajoutons que Hommel défend l’idée qu’un tel encodage est hiérarchique, et repose sur des événements qui peuvent être segmentés en plusieurs unités ayant chacune du sens.

Le lien entre perception, et planification en vue d’un but, entraîne des questionnements quant à l’influence de la capacité d’un enfant à effectuer une tâche sur sa perception d’une autre personne effectuant cette même tâche. Ainsi, dans [Somerville and Woodward, 2005] Somerville et Woodward étudient comment la capacité d’un enfant de 10 mois à résoudre une tâche de “means-end” (voir section 1.1.1) influence sa perception d’une autre personne effectuant cette même tâche, tâche qui est alors perçue comme orientée par un but [Longo and Bertenthal, 2006]. D’autres expériences montrent l’importance de la production de l’action pour la perception de l’action, notamment celles menées par [Hauf et al., 2007] (se référer à [Aschersleben, 2006] pour une étude sur le sujet, et à [Daum et al., 2008] pour une discussion sur ces résultats). Notons qu’il y figure des résultats allant dans ce sens mais avec l’utilisation d’outils, dans lesquels la seule observation de l’utilisation d’outils a des effets proches de ceux décrits dans la section 1.1.1 concernant l’espace atteignable.

Il faut noter que considérer les actions comme provenant d’une représentation anticipée des effets désirés, comme c’est le cas pour le principe idéomoteur, interroge l’action d’une manière différente que ne le font les associations stimulus-réponses, ou stimulus-récompenses.

Pour le principe idéomoteur la représentation des effets désirés implique immédiatement une planification d’actions. Pourtant l’émergence de séquences d’actions soulève des problématiques propres, et nous allons les aborder dans la partie qui suit.

1.1.4 Séquence d’actions

La planification de l’action soulève d’importantes questions concernant la manière dont une séquence d’actions unitaires significatives peut être organisée. Des travaux fondateurs [Lashley, 1951; Miller G. A., 1960] proposent une structure hiérarchique de l’action humaine. Allant dans ce sens, un modèle computationnel de schémas d’actions, organisé hiérarchiquement, et destiné à être associé à la structuration propre du domaine d’une tâche donnée (par exemple, faire le café) a été proposé par Cooper et Shallice [Cooper and Shallice, 2000] (voir figure 1.4, et [Botvinick, 2008] pour une revue).

Mais on retrouve également en neuroscience ([Dehaene and Changeux, 1997; Chersi et al., 2011; Grossberg, 1987]) ainsi qu’en robotique ([Nicolescu and Matarić, 2002; Ogino et al., 2012; Sandamirskaya and Schöner, 2010; Billing and Sandamirskaya, 2015]) des architectures pour la planification hiérarchique, et l’organisation hiérarchique du comportement. Ainsi, pour Bonini *et al.* les neurones du cortex prémoteur ventral et ceux du pariétal inférieur, avec ceux du cortex préfrontal, peuvent encoder le but d’actions intentionnelles à différents niveaux d’abs-

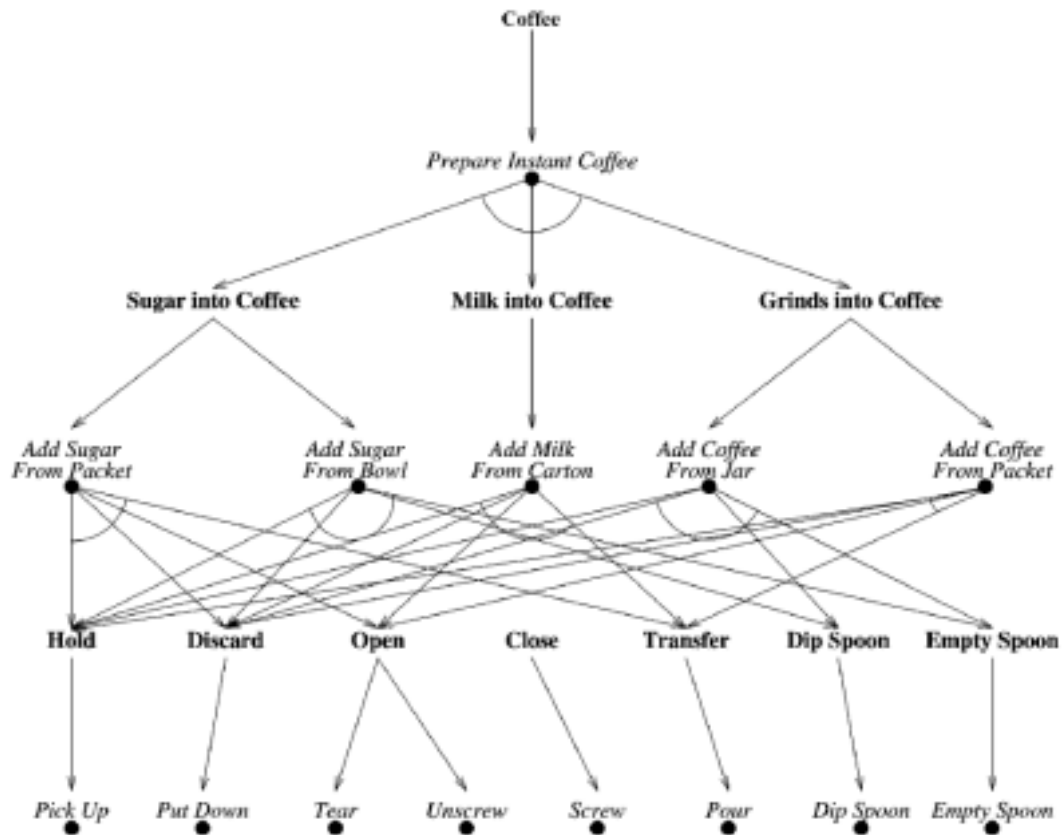


FIGURE 1.4 – Organisation des schémas (actions, en italique) et buts (en gras) dans le domaine de la préparation du café. Issu de [Cooper and Shallice, 2000]

traction motrice. De plus, le circuit pariéto-prémoteur travaille avec le cortex préfrontal pour organiser les actions motrices en séquence d’actions, et garder actif la représentation interne de l’intention motrice [Bonini et al., 2011].

D’un point de vue différent, la théorie des systèmes dynamiques montrent que ces séquences et cette organisation hiérarchique peuvent être le résultat de bifurcations dans un système dynamique plus global.

Dans [Billing et al., 2015], des champs de neurones dynamiques (DNF, voir [Schöner, 2008]) sont proposés pour combler le fossé entre les dynamiques sensorimotrices bas niveaux et des processus cognitifs plus hauts niveaux, tel que la planification. Les auteurs proposent des comportements élémentaires tels que “chercher un objet de la couleur X” ou “bouger le bras à la position Y” associés avec des DNF conditionnant l’initialisation et la finalisation (conditions de satisfaction) de ces comportements. Chaque comportement est aussi associé avec un champ de motivation (permettant de commencer le comportement) et un champ de précondition (pour savoir si le système est à même de le faire). Une fois que le champ de motivation est activé, le DNF gérant l’initialisation est aussi activé mais peut être inhibé si le champ de précondition ne satisfait pas les conditions nécessaires. Dans ce cas, ce champ peut activer le champ de motivation d’un autre comportement élémentaire qui permettra de remplir cette condition. Une fois

celle-ci effectivement satisfaite, le champ de précondition arrête l'inhibition du comportement (voir fig. 1.5).

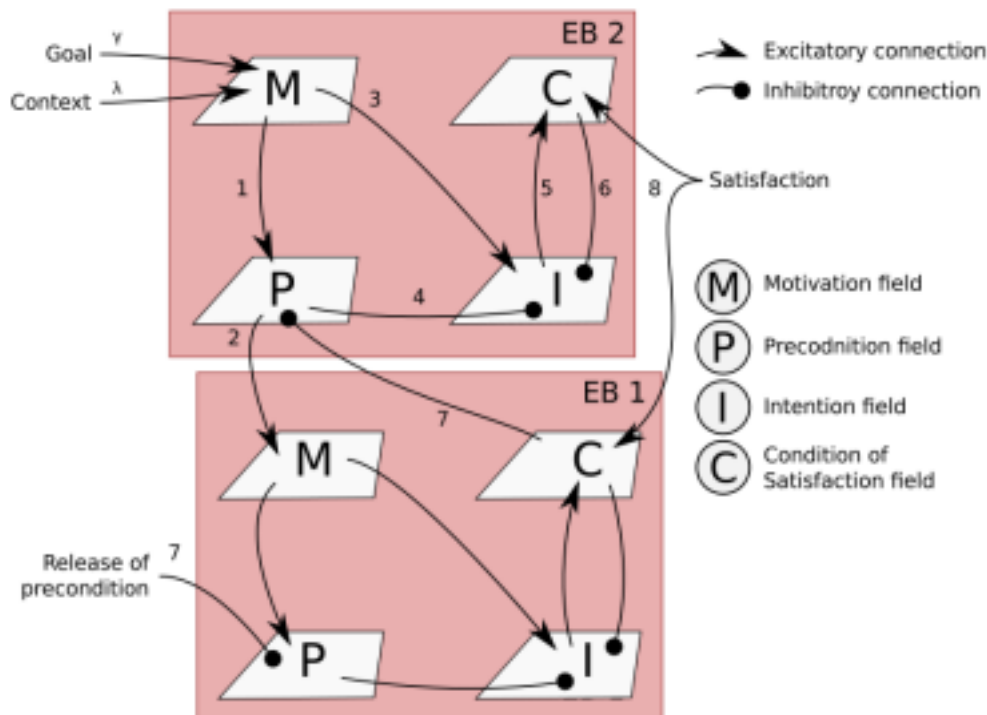


FIGURE 1.5 – Une séquence de deux comportements élémentaires EB1 et EB2. Un état but est ici représenté comme une motivation M à exécuter EB2, en ayant satisfait les préconditions P. Issu de [Billing et al., 2015].

Comme nous l'avons souligné en section 1.1.1, la capacité à faire des séquences, si elle est importante, n'est pas suffisante pour utiliser des outils. La théorie des affordances semble essentielle pour comprendre cette aptitude, et nous allons nous y intéresser dans la section qui suit.

1.1.5 Affordances, neurones miroirs et théorie de la simulation

Neurones miroirs

Les neurones miroirs ont été découverts sur les macaques, dans la zone F5 du cerveau (considérée comme la partie analogue du cortex frontal inférieur postérieur de l'humain). Ces neurones miroirs répondent lorsque des actions spécifiques sont effectuées, comme par exemple attraper un objet, mais aussi lors de l'observation d'un autre macaque ou d'un humain effectuant la même action, lorsque celle-ci est orientée par le même but. Plus précisément, ces neurones vont répondre différemment si cette même action (attraper une chose) est observée au sein de séquences motrices différentes, et est effectuée avec une intention différente (attraper cette chose *pour* la manger, ou la déplacer par exemple). Ceci semble indiquer un mécanisme en lien avec le but de l'action (voir [Rizzolatti and Craighero, 2004; Craighero et al., 2007] pour une re-

vue). Pour [Fogassi et al., 2005], les résultats suggèrent que ces neurones sont sensibles au but poursuivi tant au niveau proximal, que distal.

Notons que, en allant plus loin, pour Rolf et Asada les buts peuvent souvent être compris en terme d'affordances sous-jacentes : les buts traduisent alors un écart entre l'état présent et un état désiré imaginé [Rolf and Asada, 2015]. Les affordances décrivent dans ce cas des possibilités générales, qui peuvent être évaluées par des buts.

Affordances

Les neurones canoniques furent découverts dans la même aire (zone F5) du cerveau. Ceux-ci répondent aussi à des actions spécifiques, comme attraper un objet, mais également lors de la seule observation de ce même objet [Rizzolatti and Craighero, 2004]. Ceci semble indiquer un mécanisme d'affordance. Le concept d'affordance fut initialement introduit par Gibson [Gibson, 1979] en tant que ressources que l'environnement offre à tout animal doté de la capacité de les percevoir et de les utiliser. Notons que du point de vue de cette perspective, écologique, la perception des affordances est directe et ne sollicite pas la médiation de représentations neuronales ou mentales. Du point de vue de la perspective représentationnelle toutefois, les affordances sont internalisées par des représentations mentales afin d'être utilisées par les processus computationnels, par exemple avec un agent sensorimoteur interagissant avec le monde.

Les outils, par exemple, sont des objets manipulables générant de multiples affordances, notamment en relation avec leurs formes ou leurs fonctions [Creem-Regehr and Lee, 2005; Bub

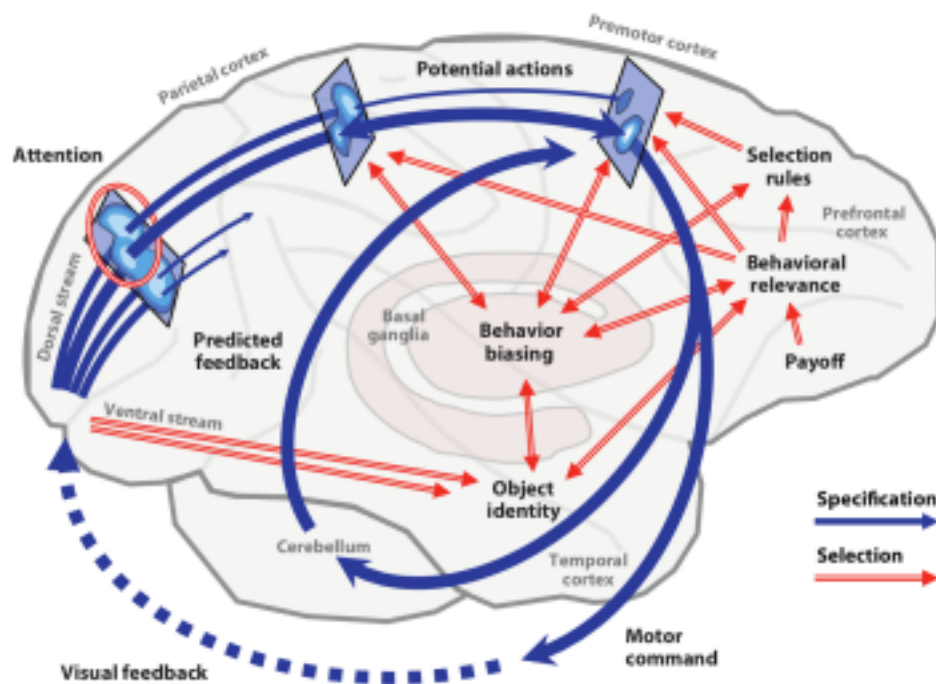


FIGURE 1.6 – Hypothèse de la compétition entre affordances, dans le contexte de mouvements guidés par la vision. Les flèches bleues et pleines illustrent les voies dédiées à la spécification de l'action, également utilisées pour guider son exécution, tandis que les flèches rouges illustrent les voies permettant la sélection de l'action. Issu de [Cisek and Kalaska, 2010].

et al., 2008]. Ellis et Tucker ont proposé le concept de micro-affordances, pour des interactions non avec l'objet complet mais avec des composantes particulières permettant l'action. Par exemple pour une taille d'objet donnée, ou son orientation particulière, il sera déclenché l'activation partielle des circuits moteurs requis pour interagir avec celui-ci [Ellis and Tucker, 2000].

Compétition entre affordances

Puisque l'environnement fourni en permanence au cerveau de nombreuses opportunités d'actions, en parallèle, cela pose le problème de leur interférence, et donc, plus généralement, d'un mécanisme de sélection.

Selon l'hypothèse de la compétition des affordances, développée par Cisek [Cisek, 2007; Cisek and Kalaska, 2010], les questions relatives à ce que l'on doit faire (sélection de l'action) et comment on doit le faire (spécification de l'action) sont traitées non successivement mais en parallèle. Le cerveau traite donc l'information sensorielle et spécifie, en parallèle, plusieurs actions potentielles disponibles, les affordances. Parallèlement, une autre voie du cerveau est supposée biaiser la sélection parmi les affordances disponibles, lesquelles sont en compétition les unes avec les autres (voir fig. 1.6). Dans de récents travaux, Thill *et al.* ont publié une étude portant sur ces mécanismes (et les modèles computationnels qui y sont liés) en insistant sur le rôle des buts, et plus spécifiquement du cortex préfrontal dans cette sélection [Thill et al., 2013].

Affordances et simulations

Les travaux de Norman [Norman, 2002] soulignent que les exemples d'affordances donnés par Gibson, à savoir percevoir le sol comme "marchable", l'eau comme "nageable" ou une chaise comme "asseable", impliquent à chaque fois une action de l'observateur, comme marcher, nager ou s'asseoir. Poursuivant cette idée, Moller *et al.* supposent que les affordances sont révélées par un processus de simulation interne des conséquences de l'action [Möller and Schenck, 2008]. En se basant sur l'information sensorielle présente, l'observateur va effectuer des simulations internes de séquence d'actions, et anticiper leurs effets. Ainsi, en opposition avec la vision écologiste de Gibson, les affordances seraient générées activement par un processus de simulation sensorimotrice, influencé par les données sensorielles, plutôt que directement obtenues par la seule analyse passive de ces données sensorielles.

Notons qu'afin d'effectuer de telles simulations, une capacité d'anticipation est requise. L'anticipation est un mécanisme fondamental du cerveau, et il est soutenu que l'apprentissage prédictif de l'information sensorimotrice joue un rôle clé dans le développement cognitif [Butz et al., 2003; Barsalou, 2009; Barsalou et al., 2007; Nagai and Asada, 2015]. L'anticipation des effets de l'action est généralement effectuée par des modèles internes (direct et inverse), comme on le verra dans la section 1.2.1.

Ainsi, dans [Möller and Schenck, 2008], les affordances sont issues de la simulation de l'action, en utilisant des modèles directs. Il sera alors possible de prédire et choisir les actions à effectuer en ce basant sur ces simulations. Celles-ci reposent sur des modèles directs (implémentées sous la forme de perceptrons multi-couches) qui prédisent des variations sensorielles basées sur l'état courant et le mouvement effectué (voir fig. 1.7). Pour des raisons de performances, le modèle de la figure 1.7 a été par la suite séparé en autant de prédicteurs différents par

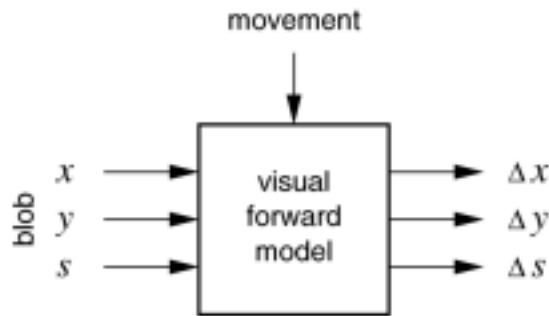


FIGURE 1.7 – Modèle direct prédisant la variation sensorielle pour la vision. Issu de [Möller and Schenck, 2008].

sensor dont la variation est prédite, chacun ayant par contre les mêmes entrées (position x et y et la taille s visuelle du blob observé).

L'idée est alors d'utiliser un processus de simulation interne en utilisant le modèle direct : en prenant l'état courant $S(t)$ en entrée, et une entrée motrice simulée $M(t)$, le modèle prédit $S(t + 1)$ en sortie, laquelle peut-être utilisée en entrée d'une autre simulation. Pour un modèle prédisant une variation visuelle et tactile, le processus de simulation interne est montré figure 1.8.

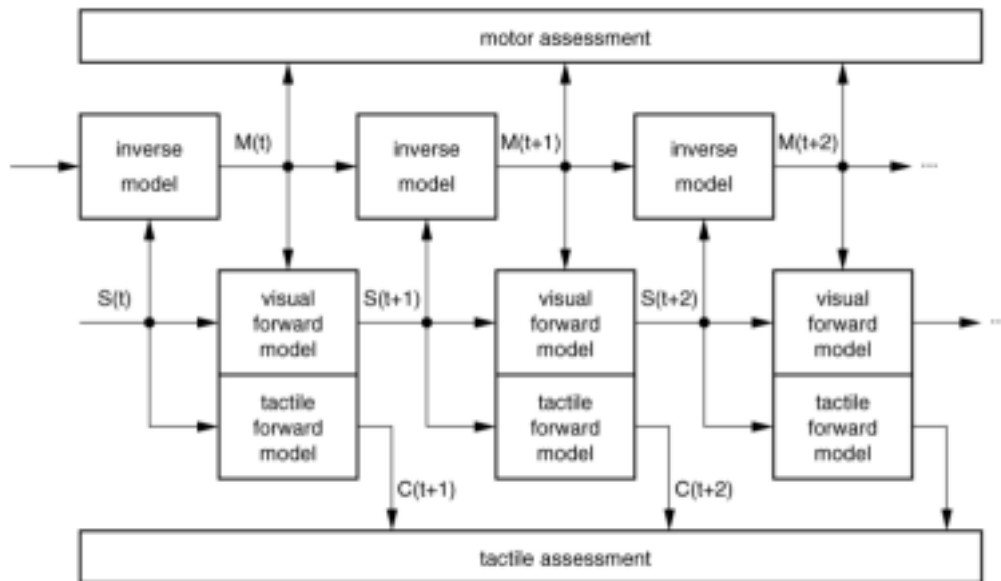


FIGURE 1.8 – Processus de simulation interne montrant l'interaction entre modèles direct et inverse. Issu de [Möller and Schenck, 2008].

Cependant, la multitude des séquences possibles étant trop riches pour être explorée, un modèle inverse est appris afin de réduire cette exploration en suggérant des actions. À l'aide d'un mécanisme d'évaluation, les actions sont alors étiquetées comme "bonne" ou "mauvaise", et le modèle inverse est alors à même de fournir un ensemble restreint de séquences prometteuses.

Contrôle et théorie de la simulation

Dans la théorie de la simulation (ou de l'émulation), il a été défendu que par la simulation mentale, une même structure neuronale pouvait sous-tendre des capacités sensorimotrices, cognitives et sociales aussi varié que la détection de soi, la distinction soi-autrui, la planification, la perception, ou encore l'imitation [Barsalou, 1999; Hesslow, 2002; Grush, 2004; Decety and Grèzes, 2006; Pezzulo et al., 2013a; Schillaci et al., 2016].

Ainsi la théorie de la simulation ne fournit pas seulement des mécanismes susceptibles de proposer une explication aux affordances ([Möller and Schenck, 2008]), mais constitue également un moyen de comprendre le lien bi-directionnel, entre la représentation des buts, et les programmes moteurs permettant de satisfaire ces buts (tel que le suppose le principe idéomoteur et la théorie du common coding).

Pour Pezzulo et Castelfranchi [Pezzulo and Castelfranchi, 2009], penser consiste ainsi à contrôler son imagination : il s'agit de la capacité à contrôler les simulations mentales afin de mettre en place des sous-but, et planifier au delà de l'ici et maintenant de la perception. Ils proposent, comme dans [Möller and Schenck, 2008] une extraction parallèle des nombreuses affordances disponibles dans l'environnement en faisant courir des simulations internes des actions possibles. Comme dans [Cisek, 2007], ils proposent également un mécanisme pour sélectionner l'une des affordances disponibles en compétition, en se basant sur leur valeur et leur faisabilité. Enfin, ils suggèrent deux mécanismes d'inhibition. L'un permet d'inhiber les nombreuses réponses automatiques, déclenchées par les affordances disponibles. Le second permet à la fois l'inhibition temporaire des commandes motrices, et l'inhibition des entrées extérieures, afin de leur substituer des stimuli générés en interne (simulés). Ils soulignent enfin l'importance d'une mémoire de travail permettant de mémoriser des buts distants ainsi que des sous-but au cours du temps (voir fig. 1.9).

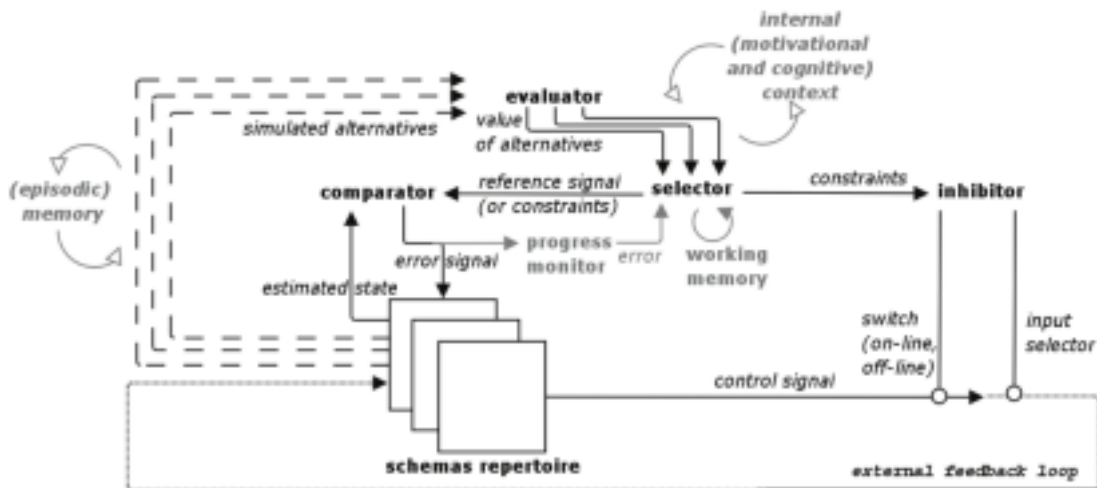


FIGURE 1.9 – Modèle théorique présenté par Pezzulo et Castelfranchi, intégrant simulations et compétitions entre affordances dans le cadre d'actions orientées vers un but. Issu de [Pezzulo and Castelfranchi, 2009].

Nous tenons ici à souligner l'importance de la distinction entre comportements orientés par un but, et les comportements habituels, ou de type stimulus-réponse. La psychologie ainsi que

la neurobiologie attestent que les uns ne sauraient se substituer aux autres et que chacun joue un rôle essentiel, irréductible à l'autre. Se pose alors la question, ouverte, de leur articulation. Afin d'illustrer comment de tels mécanismes pourraient cohabiter, nous pouvons noter l'exemple de Pezzulo *et al.* [Pezzulo *et al.*, 2013b] qui proposent un modèle, appelé Contrôleur Instrumental Mixte (MIC), permettant d'équilibrer de manière flexible chacun de ces comportements, en combinant ceux basés sur des modèles et ceux non basés sur des modèles : "model-free" pour les comportements habituels et "model-based" pour les comportements orientés par un but, voir figure 1.10.

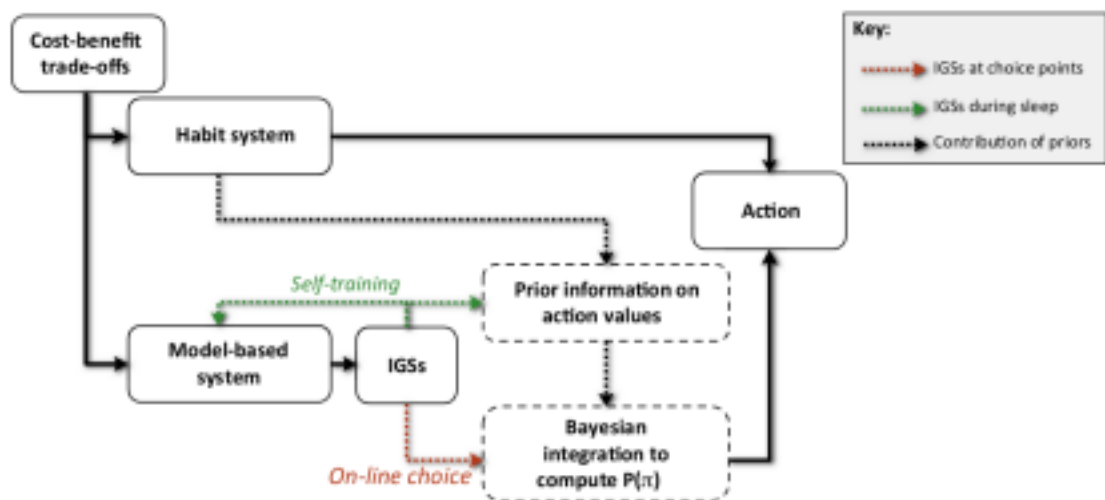


FIGURE 1.10 – Contrôleur mixte combinant, en fonction de leur coûts et bénéfices, des comportements habituels et des comportements orientés par un but (et permettant notamment de gérer des séquences - IGS). Issu de [Pezzulo *et al.*, 2014].

Après avoir étudié notre problématique sous l'angle de théories venant des sciences cognitives, de la psychologie développementale ou des neurosciences, dans la partie qui suit nous l'aborderons sous l'angle de la robotique, afin d'y observer les problématiques propres, et les types de solutions qui sont généralement proposées.

1.2 Apprentissage en robotique

Dans cette partie, qui ne vise pas à l'exhaustivité, nous commencerons par traiter la question générale de ce que peut être un apprentissage sensorimoteur en robotique (Sec. 1.2.1). Puis nous nous pencherons sur les approches les plus répandues relatives au contrôle en robotique, à savoir le contrôle optimal (Sec. 1.2.2), ainsi que l'apprentissage par renforcement (Sec. 1.2.3). À la suite d'une analyse des limitations concernant l'existence de fonctions à optimiser, nous nous intéresserons aux travaux précédents du laboratoire en considérant l'approche PerAc (Sec. 1.2.4) qui s'attache aux propriétés émergentes de l'apprentissage de la boucle perception-action. Nous proposerons alors un aperçu de la manière d'aborder la question spécifique de l'utilisation d'outils en robotique (Sec. 1.2.5). Enfin, nous regrouperons les propriétés que divers auteurs ont retenu comme essentielles pour la possibilité d'un apprentissage autonome et "open-ended" (Sec. 1.2.6).

1.2.1 Apprentissage sensorimoteur

En robotique, la problématique de l'encodage de l'information sensorimotrice peut être illustrée par la manière d'effectuer l'encodage moteur, couplé à celui de l'effecteur terminal d'un bras robotique (la main robotique). C'est un problème que l'on retrouve notamment lors de tâches consistant à atteindre un point donné ("reaching") pour, par exemple, attraper quelque chose avec la main.

Les roboticiens sont confrontés au problème de l'équivalence motrice, ou de la redondance. Ce problème apparaît pour tout effecteur doté de plus de dimensions que celles dans lesquelles sont spécifiées une cible. Ainsi, pour un bras disposant de suffisamment de degrés de liberté, il existe une infinité de trajectoires motrices permettant d'atteindre un point donné.

Pour Bullock et Grossberg [Bullock et al., 1993], il existe deux stratégies principales permettant l'encodage sensorimoteur liant les coordonnées motrices à celles de la main robotique, permettant d'effectuer des tâches de "reaching" (voir la fig. 1.11, et [Bullock et al., 1993; Braud et al., 2015] pour plus de détails).

La première stratégie consiste à apprendre la correspondance ("mapping") entre ces deux systèmes de coordonnées. Un robot peut ainsi atteindre une cible donnée X_B en activant les coordonnées motrices θ_B associées avec la position désirée de la main.

$$\theta_B = f(X_B) \tag{1.1}$$

Nous appelons cette stratégie "absolue" parce qu'une correspondance absolue est apprise entre chaque coordonnée (voir la fig. 1.11).

Les algorithmes basés sur cette stratégie peuvent par exemple considérer cet apprentissage comme étant celui d'un système homéostatique : après avoir appris ces correspondances, le système, indépendamment du bruit ou d'un quelconque changement, sera forcé d'équilibrer ses sorties sur un point d'équilibre basé sur les catégories apprises, en cas de conflit entre ses entrées visuelles et proprioceptives. Dans cette situation, les catégories apprenant cette correspondance peuvent être considérées comme des attracteurs visio-moteurs [Gaussier and Zrehen, 1995; De Rengervé et al., 2015]. Si le robot doit effectuer une séquence continue avec la main dans l'espace visuel (par exemple pour former un "8"), le robot générera alors une séquence

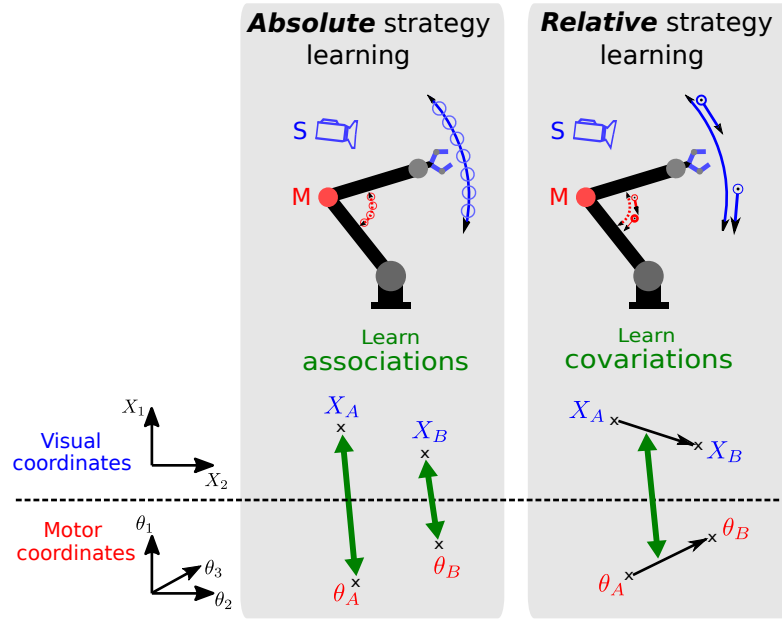


FIGURE 1.11 – Stratégie absolue et relative pour l’encodage sensorimoteur

continue de configurations motrices permettant d’effectuer cette tâche. Notons qu’avec cette stratégie, un algorithme dédié à la minimisation de la distance entre les positions courantes et désirées de la main est requis (comme un simple PID, par exemple).

La seconde stratégie, appelée “relative”, consiste à apprendre la correspondance entre chaque direction spatiale constatée par la main, et le changement correspondant dans les angles des articulations motrices qui ont provoqué ce mouvement (voir la fig. 1.11). Cette stratégie a pu faire l’objet d’implémentations neuronale, inspirées de données neurobiologiques (voir [Burnod et al., 1992, 2000]). Mais généralement, cette approche implique l’utilisation d’un modèle direct à travers la matrice jacobienne J :

$$\dot{X} = J(\theta)\dot{\theta} \quad (1.2)$$

Avec cette stratégie, un robot peut atteindre une cible non de manière immédiate, comme dans le cas de la stratégie absolue, mais à travers la direction allant de la main X_A à la cible X_B . Cette direction sera alors associée avec la variation correspondante de l’angle des articulations motrices, laquelle pourra alors être effectuée pour atteindre la cible.

$$\dot{\theta}_B = f(X_B - X_A) \quad (1.3)$$

La fonction permettant cela est généralement estimée grâce à des algorithmes d’apprentissage inverse dédiés. Généralement, les méthodes consistent à inverser la matrice jacobienne [Baillieux et al., 1984; Mussa-Ivaldi and Hogan, 1991; D’Souza et al., 2001] afin de sélectionner une unique solution, parmi la multitude des possibilités offertes, qui permettra d’effectuer la tâche.

$$\dot{\theta} = J^+(\theta)\dot{X} \quad (1.4)$$

Les modèles inverses sont bien souvent accompagnés de prédicteurs permettant d'anticiper, appelés modèles directs (voir Fig.1.12). Le modèle direct requiert une intégration des états perceptuels ainsi qu'une copie efférente des commandes motrices afin de prédire les conséquences sensorielles des actions. Le modèle inverse requiert, quant à lui, l'état actuel et désiré afin de générer les commandes motrices permettant d'atteindre l'état désiré (voir [Wolpert et al., 1998; Kawato, 1999; Wolpert and Ghahramani, 2000; Butz et al., 2003, 2006]).

Se fondant sur l'idée que le monde est modulaire, Wolpert et Kawato ont proposé un modèle basé sur des paires de modèles directs et inverses [Wolpert and Kawato, 1998], dans lesquelles les modèles directs apprennent la relation causale entre les actions et leurs conséquences, et sont ensuite utilisés comme prédicteur ou simulateur des conséquences des actions [Wolpert et al., 2003] (voir fig.1.13 extrait de [Wolpert et al., 1998]).

Cependant, une représentation modulaire du monde tient lorsqu'il y a une indépendance entre les modules, ce qui peut être notamment le cas pour des systèmes linéaires, mais le monde est tel que ses différentes parties sont en interaction dynamique non linéaire. Or définir ces modules, et leurs frontières, définit également les comportements qui leur sont adaptés, et la manière d'y résoudre des tâches. Ceci peut donc entraîner des limitations pour certaines tâches, et peut rendre difficile le passage du mono au multi-tâches.

Si les questions concernant l'encodage sensorimoteur sont centrales pour notre problématique, tant du point de vue de la psychologie (voir sections 1.1.2 et 1.1.3) que du point de vue robotique, le seul fait de travailler au contrôle d'un bras robotique afin qu'il bouge et puisse

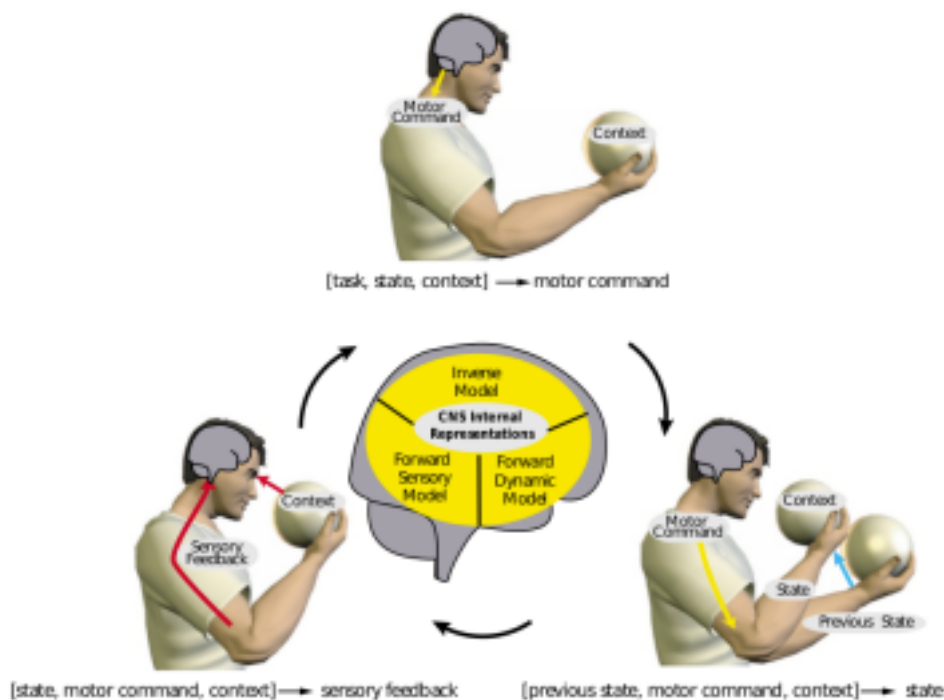


FIGURE 1.12 – Boucle sensorimotrice montrant la génération de commandes motrices proposées par le modèle inverse en haut, et en bas les modèles directs prédisant : les transitions d'états (à droite), et les retours sensoriels (à gauche) au sein du système nerveux central (CNS). Issu de [Wolpert and Ghahramani, 2000]

éventuellement effectuer une tâche, quelle qu'elle soit, soulève des questions quant à la manière d'effectuer ce contrôle et à ce qu'on attend de lui. Dans la partie qui suit, nous étudierons des approches classiques permettant d'effectuer le contrôle d'un robot pour effectuer une tâche.

1.2.2 Théorie du contrôle optimal

Dans une approche que l'on peut considérer "classique" de la robotique, l'intérêt est principalement porté sur l'accomplissement de tâches, et ce sera donc sur la manière de l'accomplir que les évaluations porteront. Ainsi, la distance à la cible, la trajectoire empruntée ou encore le temps mis à accomplir certaines tâches, étant évaluable, ont donné lieu à diverses propositions de fonctions de coûts (voir Sec. 1.2.2). L'enjeu est alors l'optimisation du comportement des robots en fonction de ces fonctions de coûts, et la théorie qui le décrit est celle du contrôle optimal. Si le cadre d'application peut être déterministe, il est également possible de l'inscrire dans le cadre stochastique, plus générique encore, et ainsi être appliqué aux Processus Décisionnel de Markov (MDP). (Voir une revue des questions y sont liées dans [Sigaud and Buffet, 2008], dont nous nous sommes ici inspirés). L'hypothèse de Markov stipule que la distribution de probabilité spécifiant un état donné ne dépend que de l'état précédent et de l'action effectuée, mais pas du passé. De manière analogue, le principe d'optimalité stipule que le choix de l'action optimale dans le futur est indépendant des actions qui ont précédé, et qui ont mené à l'état présent.

Un état est ce qui résume la situation de l'agent à chaque instant, et une action (ou décision) influence la dynamique de l'état. Il faut ajouter la notion de revenu (ou récompense) qui est associée à chacune des transitions d'état. Les MDP sont des chaînes de Markov qui visitent ces états, contrôlées par les actions et évaluées par les revenus. Résoudre un MDP revient à contrôler

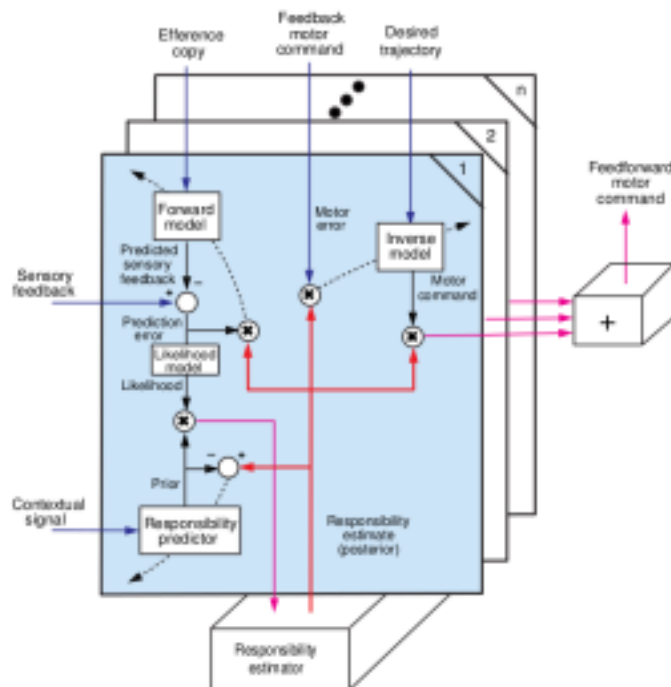


FIGURE 1.13 – Modèle de Wolpert et Kawato extrait de [Wolpert et al., 1998]

l'agent pour optimiser son comportement, c'est-à-dire en maximisant son revenu. Une solution à un MDP s'appelle une politique.

On appelle :

- S l'espace d'états dans lequel évolue le processus
- A l'espace des actions qui contrôlent la dynamique de l'état
- $T : S \times A \times S \rightarrow [0, 1]$ sont les probabilités de transition entre états. $T(s, a, s') = p(s'|a, s)$ est la probabilité d'atteindre l'état s' à partir de l'état s après exécution de l'action a
- $R : S \times A \rightarrow \mathbb{R}$ la fonction de récompense sur les transitions entre états
- $\pi : S \times A \rightarrow [0, 1]$ une politique qui code la façon dont l'agent va se comporter (la probabilité de faire une action dans un état donné).

Programmation dynamique

Une politique optimale vise à obtenir une séquence de récompenses la plus importante possible, autrement dit à maximiser le cumul des récompenses instantanées le long d'une trajectoire. Ce cumul est donné (ou estimé) à travers une fonction de valeur V :

$$\forall \pi, V^\pi : S \rightarrow \mathbb{R} \quad (1.5)$$

Par exemple, dans le cas déterministe, et en supposant N étapes, le critère fini en partant de l'état s et suivant la politique π donne la fonction de valeur suivante :

$$\forall s \in S, V_N^\pi(s) = \sum_{t=0}^{N-1} r_t \quad (1.6)$$

Le choix de la politique π s'effectue donc ainsi :

$$\pi^* = \arg \max_{\pi} (V^\pi) \quad (1.7)$$

Toujours dans le cas déterministe, en notant *suivant*(s, a) l'état dans lequel se retrouve le système en réalisant a à partir de s , on définit la politique optimale π par

$$\pi^*(s) = \arg \max_{a \in A} (r(s, a) + V(\text{suivant}(s, a))) \quad (1.8)$$

Ce maximum est susceptible d'être atteint pour plusieurs valeurs de $a \in A$, et la politique peut ne pas être unique. La fonction de valeur optimale V^* est, elle, unique pour l'état s et vérifie :

$$V^*(s) = \max_{\pi} (V^\pi) \quad (1.9)$$

$$= \max_{a \in A} (r(s, a) + V(\text{suivant}(s, a))) \quad (1.10)$$

Les équations 1.8 et 1.10 sont les équations de Bellman. Notons que ce principe a été étendu au cas continu. Les équations utilisées sont alors les équations de Hamilton-Jacobi-Bellman.

Notons également que la théorie du contrôle optimal est basée sur deux principes fondamentaux : le premier est celui de la programmation dynamique [Bellman, 1957], et le second est le principe du maximum de Pontryagin [Pontryagin et al., 1962]. Ce dernier permet de donner une condition nécessaire d’optimalité tout en évitant le problème de la malédiction de la dimension auquel la programmation dynamique est sujette [Bellman, 1957].

Toutefois, le principe du maximum de Pontryagin ne s’applique qu’au cas déterministe. Le principe d’optimalité de la programmation dynamique de son côté peut être appliqué au cas stochastique des MDP, en tenant compte d’une loi de probabilité p sur l’état d’arrivée s quand on réalise l’action a dans l’état de départ s_0 : $s = p(s|s_0, a)$. Dans ce cas, on adaptera les équations de Bellman en prenant l’espérance de la fonction de valeur. Par exemple, le cumul espéré à l’aide du critère fini dans le cas stochastique devient, avec E^π l’espérance mathématique sur l’ensemble des réalisations du MDP en suivant la politique π :

$$\forall s \in S, V_N^\pi(s) = E^\pi \left[\sum_{t=0}^{N-1} r_t | s_0 = s \right] \quad (1.11)$$

Pour finir, nous soulignerons donc que la politique choisie dépend à la fois :

- de la fonction de valeur V utilisée, et donc du critère (fini, γ - pondéré, total, moyen, *etc.*) utilisé pour cumuler l’espérance des récompenses
- de la fonction de récompenses utilisée (ou de la fonction de coût).

Intérêts et limitations de l’optimisation d’une fonction de coût

Un coût est une fonction scalaire dépendant de l’état courant du robot et de ses senseurs. Il s’agit généralement de minimiser la consommation d’énergie, de lisser la trajectoire ou encore d’optimiser la précision obtenue sur le mouvement dans le cas stochastique. Les modèles classiques de contrôle optimal en robotique s’appuient sur la minimisation de l’une de ces caractéristiques pour générer une trajectoire désirée en boucle ouverte (voir [Todorov, 2004] pour plus de détails).

Parmi ces modèles, on trouve ainsi ceux dont le but est de réduire le coût énergétique pour les muscles [Hatze and Buys, 1977]. Différents types de calcul peuvent alors être utilisés : celui-ci peut prendre en compte le couple [Nelson, 1983], considérer un modèle précis du muscle [Tani ai and Nishii, 2007; Nishii and Tani ai, 2009] ou encore chercher à minimiser les périodes de co-activation musculaire, selon le principe d’inactivation proposé dans [Berret et al., 2008; Gauthier et al., 2010].

Cependant la seule minimisation de cette énergie ne permet pas de rendre compte de la moyenne des comportements relatifs aux mouvements des bras ou des saccades oculaires, ou même de certains mouvements concernant le corps entier.

Ainsi d’autres critères peuvent être minimisés comme les secousses (*i.e.* la dérivée de l’accélération de la main) (minimum-jerk, [Hogan, 1984; Flash and Hogan, 1985]) et la dérivée du couple (minimum torque change, [Uno et al., 1989; Nakano et al., 1999]). Ce type de modèle permet d’obtenir de meilleurs résultats concernant la reproduction des dynamiques de mouvements d’atteintes constatés chez les humains, et permet également de mieux rendre compte de la relation vitesse-courbure que la loi de puissance [Lacquaniti et al., 1983].

Afin d'éviter de ne reproduire que la moyenne des résultats désirés, tout en ayant parfois des erreurs individuelles substantielles, la minimisation de la variance a également été considéré, en minimisant par exemple la variance sur la position de l'extrémité du bras à la fin d'un geste [Harris and Wolpert, 1998]. De plus, des études sur la modélisation du contrôle optimal ont également montré que la loi de Fitts (selon laquelle le temps nécessaire pour atteindre une cible dépend tant de la distance, que de la taille de la cible), peut émerger de la présence d'un bruit dont la variance augmente proportionnellement à la magnitude de la commande motrice envoyée [Tanaka et al., 2006; Guigon et al., 2008].

Notons que le coût le plus pertinent à utiliser, pour un système sensorimoteur donné, peut ne pas correspondre à notre compréhension intuitive de la tâche que nous voulons voir ce système réaliser. Par exemple, s'il semble évident que le système nerveux a de bonnes raisons de minimiser l'énergie consommée, cela apparaît moins clairement pour ce qui est d'avoir une trajectoire lissée (alors que ce coût donne de bons résultats pour reproduire la dynamique de trajectoires de mains humaines). Il semble enfin qu'il n'y ait pas de modèle qui soit le meilleur dans toutes les situations. L'optimalité semble plutôt à rechercher du côté d'une combinaison de ces différents coûts, ce qui pose le problème de la manière de pondérer chaque critère, d'autant qu'une telle combinaison est appelée à évoluer en fonction des situations.

Dans tous ces exemples effectués en boucle ouverte, la planification qui est faite est indépendante des retours des senseurs, ce qui suppose une dynamique déterministe. Ainsi, les boucles ouvertes sont efficaces pour prédire des trajectoires moyennes mais échouent à prendre en compte le bruit, les délais ou les changements imprévisibles de l'environnement. Au contraire, ceux en boucle fermée montrent de bonnes performances dans ces conditions. Ceux-ci peuvent utiliser un modèle interne permettant d'estimer l'état courant, et les performances sont alors fonction de cet estimateur, car celui-ci permettra d'anticiper les changements avant que les données sensorielles ne soient disponibles. Notons que le parallèle a été effectué avec le cerveau, qui remplit en partie les mêmes fonctions (voir [Wolpert et al., 1998]).

Que ce soit en boucle ouverte ou en boucle fermée, les fonctions de coût sont chaque fois donnée a priori, et supposent donc de pouvoir extraire objectivement les paramètres du comportement moteur qui doivent être optimisés. Il semble cependant exister des situations où les mouvements apparaissent comme suboptimaux au vu des paramètres classiquement optimisés. Dans [Becchio et al., 2008] par exemple, les auteurs observent les trajectoires de sujets devant saisir un objet et le poser sur une zone prédéfinie, ou sur la main d'un humain afin qu'il réalise ensuite un geste avec l'objet. Or le simple fait de donner un objet à un humain, au lieu de le poser sur une table, influence toute la trajectoire y compris la première partie du mouvement destiné à attraper l'objet. La trajectoire semble donc impactée par le contexte social. Cette expérience a été répliquée dans [Lewkowicz et al., 2013] avec des résultats similaires, mais Lewkowicz *et al.* montrent de plus que les cinématiques du mouvement, durant la première partie de la phase d'approche de l'objet, ont des caractéristiques suffisamment différentes pour permettre à un humain de discriminer le contexte (social ou pas). Ainsi, sans voir la seconde partie du mouvement, les expériences menées montrent un taux de succès supérieur au hasard. Notons qu'un simple perceptron (avec une couche cachée) a été capable d'apprendre à différencier les contextes à partir de ces caractéristiques. D'autres études allant dans le même sens ont également été menées, par exemple [Becchio et al., 2010; Ferri et al., 2011].

Ainsi, l'influence de l'intention sociale ou préalable apparaît à très bas niveau dans la réalisation motrice, et une optimisation semble ne pouvoir se limiter aux paramètres moteurs, en dehors de tout contexte psychologique et social.

Enfin, il faut noter que tous les modèles d'optimisation décrits jusqu'ici s'appuient sur une connaissance de la fonction de coût (ou de récompense), et sur une connaissance de la fonction de transition. Or dans le cadre d'une approche développementale, cela ne saurait être le cas. Dans la partie suivante, nous verrons les modèles permettant de palier cela.

1.2.3 Apprentissage par renforcement

L'apprentissage par renforcement (RL, [Sutton and Barto, 1998]) concerne les cas dans lesquels ni la fonction de récompense R , ni la fonction de transition T ne sont connues a priori. Or, comme nous l'avons vu avec les équations de Bellman, celles-ci sont nécessaires pour déterminer la politique optimale. L'apprentissage par renforcement offre trois manières d'apprendre une politique π :

- (i) Non basé sur un modèle (model free) : le but est de construire π sans construire T ni R ;
- (ii) Acteur - critique : cas particulier du cas (i). Le critique représente la fonction de valeur, et l'acteur la politique appliquée. Selon la valeur calculés par le critique, la tendance de l'acteur à effectuer une action va augmenter ou diminuer. La politique est alors améliorée itérativement au cours de l'expérience ;
- (iii) Basé sur un modèle (model based) : Le but est alors de construire un modèle de T et de R , et à partir de là de calculer la politique optimale (via la programmation dynamique).

Parmi les méthodes non basée sur un modèle, on peut distinguer celle de Monte Carlo ainsi que celles de différence temporelle, qui sont les plus caractéristiques de l'apprentissage par renforcement.

Dans les méthodes de Monte Carlo, le but est d'estimer $V(s)$. La méthode consiste à simuler un grand nombre de trajectoires à partir de chaque état s , et à estimer $V(s)$ en moyennant les coûts observés sur ces trajectoires. Afin de pouvoir effectuer ces simulations, les trajectoires doivent être finies. Dans ces méthodes, il faut que l'expérience soit finie pour qu'il y ait un apprentissage : elles ne sont donc pas incrémentales.

Cependant la fonction de valeur peut être mise à jour à la suite de chaque transition, et c'est ce principe qui est à la base des méthodes de différence temporelle (sorte de version "on-line" des méthodes de Monte Cristo). Pour ce faire, au lieu de calculer la fonction de valeur à partir du cumul total espéré, on l'estime à chaque transition en calculant l'erreur par rapport à l'itération précédente.

L'algorithme TD(0) dit de « différence temporelle » compare la récompense obtenue avec celle estimée. Considérons, comme la majeure partie de la littérature sur le sujet, le cas du critère γ -pondéré suivant : $E[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots + \gamma^t r_t + \dots | s_0]$. Pour une fonction de valeur basée sur un tel critère, on aurait une mise à jour du type :

$$\mathbf{TD(0)} : V(s_t) \leftarrow V(s_t) + \alpha[r_t + \gamma V(s_{t+1}) - V(s_t)] \quad (1.12)$$

Si cet algorithme permet d'estimer la fonction de valeur (et la preuve de sa convergence a été apportée [Dayan and Sejnowski, 1994]), la fonction de transition reste inconnue. De fait, étant dans un état, l'algorithme ne permet pas de prédire quelle action maximisera la récompense, ce n'est qu'une fois dans l'état qu'il pourra l'estimer.

Afin de remédier à cela, une fonction de valeur Q basée sur les couples (s, a) a été proposée [Watkins, 1989]. $Q^\pi(s, a)$ est la valeur espérée pour le processus partant de s , exécutant l'action a , puis suivant la politique π . Notons alors que $V^\pi(x) = Q^\pi(x, \pi(x))$. Dans les mêmes conditions, l'algorithme SARSA propose l'équation de mise à jour suivante :

$$\text{SARSA : } Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (1.13)$$

Si sa preuve de convergence a également été apportée [Singh et al., 2000], l'inconvénient majeur de cette méthode vient du fait que la mise à jour ne peut se faire que pour l'action réalisée effectivement (du fait de la présence de a_{t+1}), et donc l'apprentissage ne peut que suivre la politique courante ("on-policy"), ce qui peut notamment poser des problèmes quant à l'exploration de nouvelles trajectoires.

Afin de permettre une mise à jour en fonction de l'action optimale, indépendamment de l'action réelle effectuée, et donc de la politique suivie ("off-policy"), une mise à jour modifiée a été proposée et constitue l'algorithme de Q-learning [Watkins and Dayan, 1992]. L'équation substitue l'action optimale (présence de max_a) à l'action a_{t+1} réellement effectuée, et devient alors :

$$\text{Q-learning : } Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (1.14)$$

Enfin, il faut noter que pour pouvoir apprendre autre part qu'à l'état visité par l'agent, et donc accélérer l'apprentissage, il est possible de stocker une mémoire des transitions parcourues et donc de propager l'apprentissage à ces valeurs durant l'apprentissage. Une telle mémoire s'appelle trace d'éligibilité, et a donné lieu aux algorithmes TD(λ), SARSA(λ) et Q(λ). Pour une étude plus approfondie sur la théorie, les applications des processus décisionnels de Markov en intelligence artificielle, et sur l'apprentissage par renforcement, voir les travaux de [Sigaud and Buffet, 2008] dont nous nous sommes inspirés.

Les approches visant à maximiser (ou minimiser) une récompense (ou un coût) sont sujettes aux limitations évoquées dans la partie précédente. Toutefois, celles-ci peuvent être en partie remplacées, ou bien accompagnées d'autres approches complémentaires. Notamment, certaines approches peuvent se concentrer sur la boucle des perceptions et actions propre à l'interaction d'un robot dans son environnement. De part ses propriétés mécaniques, ou encore au vu de la manière qu'aura un robot d'encoder l'information sensorielle, de ses réflexes (issue de l'observation animale ou humaine), ou encore de l'interaction sociale, de nombreuses propriétés peuvent émerger d'apprentissages prenant en compte ces nombreuses dimensions, et peuvent parfois donner lieu à des comportements qu'un observateur extérieur pourrait qualifier de haut niveau. Dans la partie suivante, nous examinerons l'un de ces modèles développés dans le laboratoire où cette thèse a été effectuée, et qui a fortement inspiré notre démarche.

1.2.4 Approche PerAc

Le principe clé du modèle Perception-Action (PerAc) [Gaussier and Zrehen, 1995] repose sur l'aspect déterminant de l'interaction entre un robot et son environnement. Plus précisément, il

met en œuvre l'idée que la création d'une représentation sensorielle n'est possible qu'au regard des actions qui en découlent. Il propose un conditionnement réciproque des actions et des perceptions, permettant l'émergence de comportements cohérents avec les besoins du robot et son environnement physique et social.

Le modèle proprement dit est un modèle générique de couplage entre des sensations et des actions permettant de construire des comportements. Inspiré de la counter propagation [Hecht-Nielsen, 1987], PerAc propose le conditionnement d'actions (celui d'une direction à suivre, ou encore d'une configuration motrice par exemple) à partir de l'information sensorielle (la reconnaissance d'un lieu spatial, d'une position particulière d'un membre, d'une expression faciale, *etc.*). La boucle sensorimotrice est refermée par l'environnement, comme le montre la Figure 1.14.

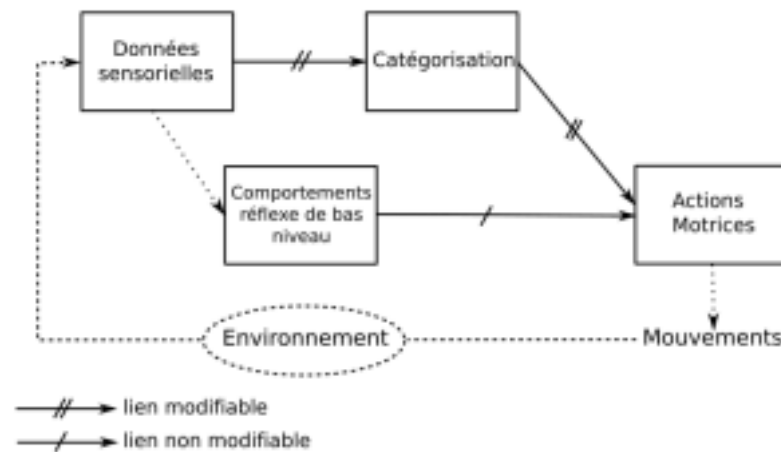


FIGURE 1.14 – Boucle sensorimotrice de l'architecture PerAc. Issu de [Boucenna, 2011].

Cette architecture bas niveau permet d'obtenir des comportements sans avoir recours à une représentation symbolique du but. En prenant ses entrées directement dans les capteurs et sans caractérisation symbolique de la situation, le modèle évite le problème de l'ancrage des symboles de l'intelligence artificielle classique ("symbol grounding problem", [Harnad, 1990]).

Une voie réflexe de bas niveau génère des comportements simples (comme ceux liés à la survie) : à partir d'informations sensorielles frustrées (infrarouges, tactiles, vision de bas niveau...), des comportements *ad-hoc* comme l'évitement d'obstacles peuvent être générés. Ces comportements peuvent être appris, et ainsi, à partir de perceptions potentiellement de plus haut niveau, des comportements peuvent être anticipés. Cette dynamique permet l'émergence de comportements plus complexes comme la perception d'objet [Maillard et al., 2005] ou le retour vers une source [Giovannangeli and Gaussier, 2008].

Dans le cadre de l'apprentissage d'une tâche de navigation supervisée, l'humain peut donner la direction à suivre comme signal désiré. Le robot apprendra alors à associer l'état courant à la direction désirée. A partir de 3 états (ou plus) dont les actions associées pointent vers un but virtuel, il est possible de construire un bassin d'attraction autour de ce but. La dynamique de reconnaissance des lieux et des actions permet alors au robot de converger naturellement vers le but, sans que celui-ci ne soit explicitement codé [Gaussier and Zrehen, 1995; Gaussier et al., 2000]. Elles peuvent également fournir une stratégie de homing si celles-ci font converger vers

une trajectoire (voir fig. 1.15). L'idée principale ici est que les trajectoires ne sont pas prescrites mais émergent de la dynamique sensorimotrice.

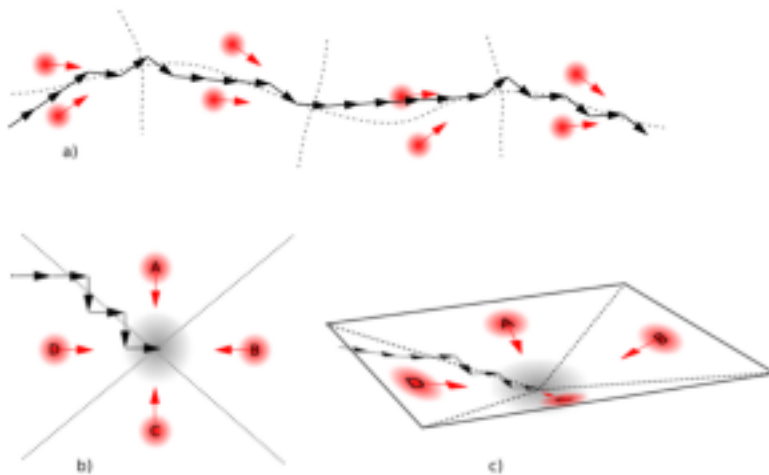


FIGURE 1.15 – Chaque lieu est associé à une direction à suivre (flèche rouge). Chaque fois que le robot se trouve dans un lieu, il se tourne automatiquement dans la direction apprise dans ce même lieu. a) représente une trajectoire formée d'associations lieu-action. b) et c) montrent l'apprentissage d'un lieu but avec des associations lieu-action, ainsi que le gradient de convergence que peut suivre un robot pour y arriver. Issu de [Jaffret, 2014].

Dans de grands environnements, ou bien lorsque ceux-ci sont très changeants, de telles stratégies montrent leurs limites. Il n'est notamment pas possible à l'apprentissage de s'adapter pour trouver d'autres chemins, ni de choisir entre plusieurs chemins possibles pour rejoindre un lieu (voir fig. 1.16), sauf à créer des bassins d'attraction différents pour chaque but [Gaussier et al., 2000].

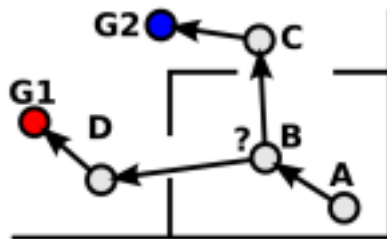


FIGURE 1.16 – Selon le but, deux chemins différents peuvent être choisis. Ceci illustre la nécessité d'une planification différente de celle provenant des seules associations lieu-action. Issu de [Hirel, 2011]

Afin de résoudre ce genre de problèmes, une solution consiste à mettre en place des mécanismes de frustration qui détecteront une stagnation dans les progrès du robot à réaliser sa tâche, ce qui permet d'entraîner un changement de stratégie lorsque la frustration est trop grande [Hasson and Gaussier, 2010; Jaffret, 2014]. Une autre solution consiste à utiliser des processus de planification de trajectoires et de sélection d'action, comme ceux basés sur la construction d'une carte cognitive [Banquet et al., 1997].

L'idée de carte cognitive a été émise par Tolman [Tolman, 1948] à partir d'expériences dans lesquelles des rats traversent un labyrinthe. Une carte cognitive correspondrait à une représentation interne de l'environnement permettant à l'animal de trouver des chemins en prenant des raccourcis et en évitant les obstacles. Les cartes cognitives permettent aussi d'expliquer par un apprentissage latent la vitesse à laquelle le comportement du rat s'adapte lorsque une récompense (nourriture) commence à être distribuée. En effet, une telle modification du comportement peut difficilement être expliquée par des apprentissages stimulus-réponse classiques.

La découverte de "cellules de lieu" [O'Keefe and Nadel, 1979], c'est-à-dire des cellules répondant à des endroits précis de l'espace, a permis une meilleure compréhension du fonctionnement de la navigation spatiale chez l'animal. Il a de fait été proposé que la carte cognitive soit présente dans l'hippocampe. La discrétisation de l'espace en cellules discrètes permet, par ailleurs, d'appliquer des modèles probabilistes de sélection de l'action, comme des PDM. Une carte cognitive correspondrait alors à une représentation interne de l'environnement permettant à l'agent d'inférer son chemin parmi plusieurs. Les lieux sont reliés entre eux par des connexions synaptiques, et forment ainsi une carte topologique de l'environnement.

Toutefois, la question se pose alors de savoir ce qui doit être appris. Car si la carte permet de sélectionner le prochain lieu à atteindre, le système ne saurait pour autant déduire l'action à effectuer, puisque celle-ci dépend également de la situation initiale. La solution proposée par [Banquet et al., 1997] consiste à apprendre les transitions en lieux. La topologie étant insuffisante à rendre compte de comportements basés sur la motivation, un mécanisme permet, lorsqu'un but est découvert, de renforcer la connexion entre le neurone qui code la motivation pour rejoindre ce but, et le neurone représentant le lieu courant. Par la suite, l'activité de la motivation se propage à travers la carte cognitive en diminuant à chaque connexion synaptique. L'activité d'un neurone, dans ce modèle, est calculée comme la valeur maximale des activités post-synaptiques qu'il reçoit. Plus le lieu est proche du but plus son activité dans la carte cognitive est grande, et on peut alors trouver le chemin optimal vers le but par une simple remontée de gradient. Ce système est similaire à l'algorithme de Bellman-Ford, calculant le chemin le plus court dans un graphe. De cette manière le robot est à même de remettre en cause la planification de son chemin lors de l'apparition d'un raccourci, ou encore sélectionner le chemin optimal vers un but particulier. Ce modèle a été implémenté et utilisé dans des tâches de navigation robotique [Gaussier et al., 2002; Banquet et al., 2005; Cuperlier et al., 2007; Hirel et al., 2011].

La carte cognitive est donc construite et encodée sous la forme d'un graphe dont les nœuds sont les transitions entre les cellules de lieu, et qui permet de planifier des déplacements dans cet espace topologique. Il est construit en ligne, et la granularité avec laquelle l'espace est pavé par les cellules de lieu peut être contrôlée par un degré de vigilance.

Cette architecture a par la suite été adaptée au contrôle d'un bras robotique [Rolland de Rengerve, 2013]. La configuration motrice du robot, qui lui est donnée par les informations proprioceptives, sera l'espace dans lequel des "cellules de lieu" encoderont certaines configurations. Afin de se déplacer dans cet espace, il ne sera plus question d'orientation du robot, généralement dans l'espace 2D, mais du contrôle du bras dans l'espace de ses degrés de libertés (DDL). Aux associations entre la perception visuelle de la main du robot, et la configuration proprioceptive correspondante, il a été ajouté un modèle du muscle pour contrôler le bras d'une configuration proprioceptive à une autre [De Rengervé et al., 2015]. De plus, ces associations visio-motrices bas niveau permettent, par le maintien de leurs homéostat, des comportements d'imitation immédiate ou différée [de Rengervé et al., 2010; Rolland de Rengerve, 2013]. Plus de détails seront

donnés dans la section 2.1. Mais en reliant ces états, et en apprenant leurs transitions, il est par la suite possible d'obtenir une carte cognitive semblable à celles développées pour la navigation et permettant des comportements motivés. Dans [De Rengervé et al., 2011] une telle carte est par exemple utilisée pour une tâche de "tri de canette", dans laquelle une canette sera déposée à un endroit ou à un autre selon sa couleur. Il en résulte donc une architecture, figure 1.17, qui part de l'encodage bas niveau d'attracteurs sensorimoteurs, et qui permet par la suite de créer une carte cognitive correspondant à des comportements dit plus haut niveau, comme la planification et l'action orientée vers un but. Les comportements pourront être biaisés par une interaction avec le robot, par exemple un signal négatif pourra entraîner un changement du but poursuivi par le robot [De Rengervé et al., 2012].

Les théories décrites jusqu'ici ne sont pas dédiées à l'utilisation d'outils. Afin d'appréhender les spécificités de cette problématique, nous allons, dans la section suivante, proposer un aperçu de différentes approches permettant d'obtenir, ou d'exploiter, la capacité à utiliser des outils.

1.2.5 Utilisation d'outils en robotique

Selon Beck [Beck, 1980], l'utilisation d'un outil consiste en l'emploi d'un objet externe de l'environnement, pour altérer efficacement la forme, la position ou l'état d'un autre objet, d'un autre organisme, ou encore de l'utilisateur lui-même.

Dans cette définition répandue de ce qu'est un outil, il est intéressant de noter qu'un outil est par définition extérieur au corps de l'utilisateur. De fait, la problématique de l'utilisation d'outil

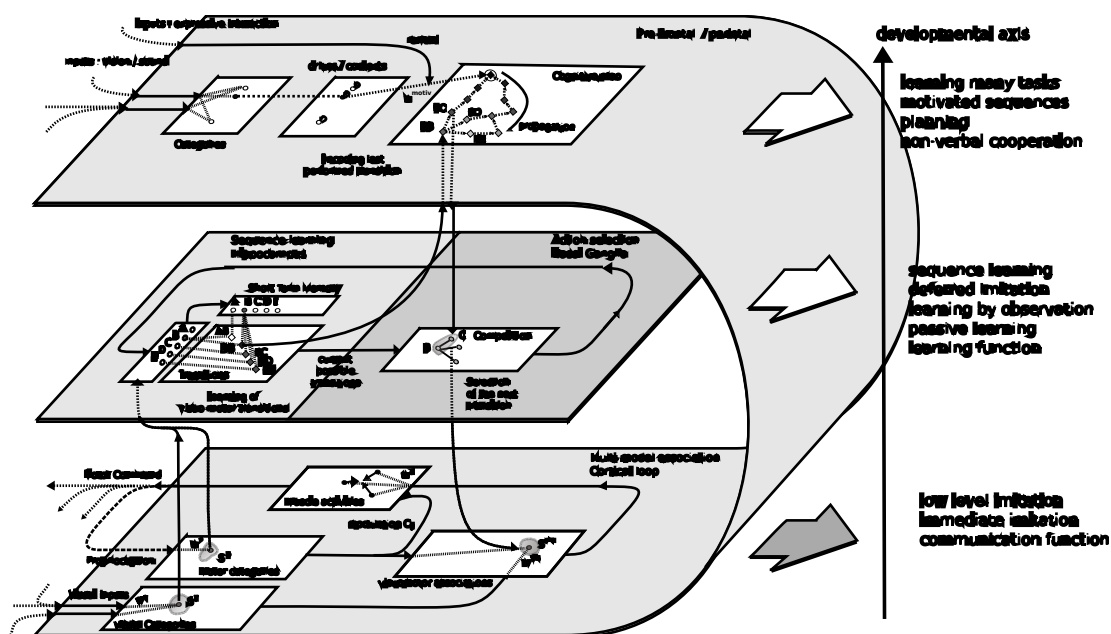


FIGURE 1.17 – Modèle issu des travaux de [Rolland de Rengerve, 2013] portant la carte cognitive sur un bras robotique. L'encodage d'associations visuomotrices, que l'on voit en bas de la figure, sert à l'étage supérieur à encoder des transitions entre ces associations, lesquelles, au niveau supérieur, sont utilisées par la carte cognitive qui vient biaiser le choix de ces transitions en fonction de buts.

aura toujours à traiter, à la fois, la question de ce qui constitue le corps, dans sa relation avec la question de l'adaptation à l'environnement.

Du seul point de vue de l'aide que des robots peuvent apporter à des humains, il est à noter que celle-ci serait plus simple à mettre en place si les robots sont capables de s'adapter à un environnement humain, c'est-à-dire fait pour l'homme et non pour la machine. Afin d'assister un humain, un robot devrait donc idéalement être capable de se saisir de la diversité des outils faits pour l'homme, plutôt que d'avoir un outil qui fasse déjà parti de son corps, et donc déjà attaché à son effecteur terminal (voir [Kemp and Edsinger, 2006; Hoffmann et al., 2014]).

Ainsi, comme nous l'avons évoqué plus haut (sec. 1.1.1), l'utilisation d'outil se rapporte à l'adaptation du schéma corporel. Hoffmann et collègues proposent dans [Hoffmann et al., 2010] une revue complète de la question de l'adaptation du schéma corporel lors de l'utilisation d'outils, en robotique.

L'utilisation d'outils soulève également le problème de l'adaptation cinématique. Dans [Rolf et al., 2010], un seul réseau neuronal récurrent est utilisé pour apprendre la cinématique inverse des différents outils, sans réapprentissage ou oubli du schéma corporel. Cependant, la longueur de l'outil est un paramètre de ce réseau neuronal, qui doit être connu à l'avance. Considérant ce problème, dans [Jamone et al., 2012] Jamone et collègues montrent comment la cinématique pourrait être adaptée à l'utilisation de l'outil quand aucune information n'est donnée sur l'outil saisi. Ils ont utilisé pour cela le modèle IMLE qui permet l'apprentissage incrémental de différentes solutions lors d'entrées similaires, et permet ainsi de représenter les relations d'un vecteur d'entrée à un vecteur de sortie dans différents contextes inconnus spécifiques. Pendant l'apprentissage et la phase d'utilisation, l'algorithme traite automatiquement d'un ensemble approprié de modèles linéaires locaux, et cette connaissance est exploitée pour une tâche d'atteinte.

Cependant, Nabeshima et collègues [Nabeshima et al., 2006] défendent l'idée que le véritable problème pourrait être plutôt celui de la détection des altérations corporelles, basées sur les données sensorielles. Ils proposent ainsi un modèle permettant d'adapter le schéma corporel du robot à un outil. Le robot (un bras à 2 degrés de liberté dans le plan, et une caméra) apprend les associations entre l'intégration spatiale et temporelle des informations visuelle et tactile, lorsque la main du robot touche une cible. L'association est ensuite stockée dans une mémoire associative. De cette manière, lorsque le robot touche la cible avec cette fois un outil (un bâton), une nouvelle association entre les informations visuelles et tactiles est apprise dans la mémoire associative, et le contrôleur cinématique du bras est adapté.

La question du schéma corporel peut également être vue comme une étape nécessaire permettant d'accéder à celle d'après, et qui consiste en l'apprentissage des affordances et de leur utilisation. C'est notamment le cas de Fitzpatrick et collègues, qui dans [Fitzpatrick et al., 2003] proposent qu'un robot, après avoir appris à distinguer son corps du reste du monde, apprenne ses affordances à travers les conséquences de ses actions, afin de pouvoir par la suite utiliser ces affordances pour agir en vue d'un but. Notons que ces auteurs insistent sur le fait qu'il est difficile de savoir quand une perception se termine et qu'une action commence, et que la distinction entre les domaines moteur et sensoriel est floue. Ceci se traduit, dans leurs expériences, par le fait que la similarité de deux actions ne soit pas considérée du point de vue de la cinématique, issue des moteurs, mais du point de vue des effets sur le monde extérieur. Cette propriété est alors utilisée pour la compréhension, et l'imitation d'actions effectuées par d'autres.

Ainsi, si l'adaptation du schéma corporel à un outil peut rendre l'utilisateur compétent, dans son usage de cet outil, l'exploitation des affordances peut, elle, fournir un premier pas vers un

usage planifié d'outils. Dans [Stoytchev, 2005], Stoytchev propose qu'un robot apprenne des associations entre, d'une part, un outil et un ensemble prédéfini de comportements, et, d'autre part, un effet, en l'occurrence l'invariant d'un ensemble d'observations. Par la suite, ces associations sont utilisées afin de créer dynamiquement une séquence de comportements permettant de résoudre une tâche.

D'autres travaux reprennent l'idée consistant à utiliser l'apprentissage d'affordances, y compris celles provenant d'outils, afin d'obtenir un but donné, grâce à la connaissance des effets que chaque action pourra, selon le contexte, donner. Notons la similarité de la mécanique à l'œuvre ici avec les schémas, ou schèmes sensorimoteurs de Piaget (voir [Piaget and Cook, 1952]). Dans les deux cas, il s'agit de l'apprentissage d'un contexte dans lequel une certaine action pourra produire un certain résultat. Dans la vision constructiviste de Piaget, l'enfant apprend graduellement, à mesure qu'il interagit avec l'environnement, à coordonner ensemble des schémas bas et hauts niveaux. Comme nous l'avons souligné plus haut (sec. 1.1.1), en soulevant tout autant la question de l'encodage sensorimoteur que celle de la capacité à planifier, la problématique de l'utilisation d'outils soulève un large spectre de questions fondamentales à la robotique développementale (voir [Guerin et al., 2013]).

Une revue des travaux consistant à utiliser les affordances pour l'utilisation d'outils, et permettant une planification, est faite dans [Ugur et al., 2011] (voir aussi les travaux de [Tikhanoff et al., 2013]). Ugur et collègues y classent les différents apprentissages (du type contexte/action/résultat) selon ce qui est appris, et la méthode d'apprentissage utilisée. Dans leurs travaux, ils se positionnent, quant à eux, dans la lignée du principe idéomoteur. En effet, en suivant ce principe, Elsner et Hommel proposent que deux phases soient requises pour l'acquisition de comportements orientés par un but [Elsner and Hommel, 2001]. Dans une première phase, les actions et leurs effets sont associés de manière bi-directionnelle lors d'une phase de babillage moteur. La seconde phase consiste en l'obtention d'un contrôle intentionnel des actions, par la prédiction des effets que ceux-ci peuvent créer.

S'inspirant de ces travaux, Ugur et collègues proposent (dans [Ugur et al., 2011]) une approche également en deux phases. Dans la première, une exploration sans but permet à un robot de découvrir les affordances offertes par l'environnement. Elles consistent en l'association bi-directionnelle entre un comportement donné (parmi un répertoire pré-codé de comportements discrets), et un effet, lequel est défini comme la différence entre les caractéristiques (vecteurs de valeurs continues) initiales et finales d'un objet. Dans un second temps, le robot contrôle ses actions par la prédiction de leurs effets. Pour cela, les auteurs tirent parti du fait que les effets et les objets (qui font contexte) soient dans le même espace. Il est de fait possible d'ajouter un effet à l'état en cours, et ainsi, de créer un arbre de recherche dont les noeuds sont composés d'états perceptuels, et les arêtes sont composées de paires comportement-objet. Lorsque le robot a un but donné, il crée un tel arbre, et l'étend graduellement, en partant de l'état qui est à la distance minimale du but. Cela permet au robot de sélectionner une séquence, au sein de cet arbre, lui permettant d'atteindre l'état satisfaisant l'objectif.

Dans [Forestier and Oudeyer, 2016], les auteurs proposent un modèle dans lequel l'espace des senseurs est associé à celui des paramètres permettant le contrôle des moteurs, lequel est effectué à l'aide de DMP (*Dynamical Movement Primitives*, voir [Ijspeert et al., 2013]). Ils proposent un environnement simulé à deux dimensions dans lequel est présent un bras robotique à 4 degrés de libertés, deux outils (des batons de tailles différentes), un objet et des cibles. L'apprentissage consiste à associer les trajectoires motrices, issues des paramètres moteurs, aux résultats

obtenus dans l'espace des senseurs. Ceux-ci sont constitués par les trajectoires de l'effecteur terminal du bras robotique, celles de l'extrémité de chaque outil, et enfin la position finale de l'objet et sa distance à la cible la plus proche. Les auteurs comparent ensuite la progression des résultats lors de tâches consistant à atteindre un but, dans l'espace des senseurs.

Les auteurs testent deux aspects qu'ils estiment être centraux : le premier est la motivation intrinsèque, déterminant l'exploration effectuée, le second est la manière dont est représentée l'encodage sensorimoteur. Pour traiter ce second aspect, ils opposent deux types d'architectures : celle dite plate, et celle dite hiérarchique. Dans l'architecture plate, l'ensemble des senseurs est réuni en un tout, quand dans la seconde architecture cet ensemble est séparé en sous-ensembles, lesquels sont reliés par des liens hiérarchiques. Ils montrent dans leurs travaux que les architectures plates apprennent moins efficacement lorsque l'espace des senseurs a trop de dimensions, comparativement à l'apprentissage hiérarchique, qui est donc privilégié. De la sorte, la question de la motivation intrinsèque se retrouve aussi concernée par le choix du sous-espace, hiérarchiquement situé, de senseurs dont il faut privilégier l'exploration. Par ailleurs, l'importance de cette motivation est d'autant mieux illustrée, dans cet exemple d'utilisation d'outils, que des progrès dans une tâche haut niveau ne sont possibles que si des progrès dans les niveaux inférieurs ont déjà été effectués.

L'apprentissage autonome d'une telle hiérarchisation sensorimotrice n'est pas l'objet d'étude des auteurs, même si ces derniers soulignent l'importance d'un tel apprentissage. Ajoutons que l'on peut se demander si une telle hiérarchisation doit être apprise, en tant que telle, ou bien si celle-ci ne doit pas être émergente, et n'apparaître qu'en tant que solution particulière à un problème particulier. Si l'on part du principe qu'un outil ne devient tel que grâce aux actions qui y sont liées (ses affordances), alors lorsque deux objets sont dans l'environnement, il est, d'un point de vue théorique, tout autant possible que le premier devienne un outil permettant une action sur le second, que l'inverse.

La plupart des résultats décrits jusque maintenant n'ont pas pour objectif premier l'autonomie de développement du robot, ou encore la capacité à résoudre de nouvelles tâches en se basant sur une exploitation nouvelle de précédents apprentissages, même s'il est possible d'exploiter ces méthodes dans un tel objectif. Puisque ces capacités sont liées à notre problématique générale, décrite dans l'introduction, dans la partie suivante nous extrairons plus spécifiquement de la littérature les quelques propriétés générales considérées comme importantes pour obtenir de tels apprentissages.

1.2.6 Apprentissage autonome et "open-ended"

La question d'un apprentissage ouvert à différentes finalités, afin de s'adapter aux variations du monde, du robot lui-même et des buts qui l'animent, est une question fondamentale en robotique développementale et autonome. Les nombreux travaux ([Lungarella et al., 2003; Prince et al., 2005; Oudeyer et al., 2007; Stoytchev, 2009; Asada et al., 2009; Law et al., 2014]) portant sur cet apprentissage (dit "open-ended") s'inspirent principalement d'études portant sur le développement de l'enfant, et ont pour objectif de proposer des mécanismes généraux d'apprentissage capables d'intégrer des expériences, de potentiellement très longue durée, au sein d'un ensemble cohérent de compétences, et qui pourront être réutilisées de différentes manières.

D'une part, il faut noter que le développement de l'enfant s'accompagne de changements radicaux et systématiques de son expérience multimodale, et des corrélations sensorielles qui

l'accompagnent. Ceci a mené à l'idée que le développement constitue une extension continue des compétences existantes, afin de tenir compte des nouvelles expériences perceptuelles [Smith and Gasser, 2005], ou de l'évolution des dynamiques complexes entre le corps et l'environnement ([Smith and Thelen, 2003]) auxquelles le cerveau de l'enfant doit s'adapter.

En allant plus loin, il a été soutenu que les entrées sensorielles sont initialement non spécifiées, ou "étiquetées", et que le rôle spécifique de chaque modalité doit être inconnu *a priori*. Cette question fondamentale a été illustrée par Denett ([Dennett, 1978]) de la manière suivante : il nous invite à nous imaginer emprisonné dans la salle de contrôle, sans fenêtre, d'un robot géant. Les murs sont recouverts de senseurs, sous la forme de lampes, et d'effecteurs, sans qu'aucun d'entre eux ne soient étiquetés. De plus, peu importe de savoir si derrière le déclenchement d'un effecteur toute une machinerie, potentiellement complexe, se met en branle. Le but final est non seulement de les étiqueter de manière pertinente, mais surtout de contrôler le répertoire de comportements du robot, pour lui permettre par exemple de marcher.

Pierce et Kuipers se sont intéressés à un agent apprenant équipé d'un appareil sensorimoteur "non étiqueté" [Pierce and Kuipers, 1997] : l'agent n'a aucune connaissance *a priori* du système moteur ou des senseurs. L'appareil sensorimoteur se résume alors à des vecteurs de senseurs et de moteurs de données brutes. Les auteurs soutiennent que ces données brutes ne sont pas adaptées pour décrire la structure du monde et prédire les effets de l'action sur les senseurs, et nécessitent donc une étape d'abstraction. Cette abstraction pourra être obtenue en générant des caractéristiques (par exemple en utilisant des ACP), en supposant des relations approximativement linéaires entre les intensités motrices et la dérivé des caractéristiques extraites [Pierce and Kuipers, 1997]. Une hypothèse similaire a été explorée par [Stronger and Stone, 2006], ou encore dans le cadre de la théorie de l'information [Olsson et al., 2006; Lungarella and Sporns, 2006; Kaplan and Hafner, 2005].

D'un autre côté, savoir dans quelle mesure les compétences bas niveaux (comme les associations sensorimotrices) et les capacités dites hauts niveaux (comme la planification de l'action, ou l'utilisation d'outils) partagent des mécanismes d'apprentissage communs demeure une question ouverte. Les dynamiques internes de l'agent et celles de l'environnement, couplées durant l'acquisition de capacités sensorimotrices, pourraient permettre à ces compétences de se développer [Smith and Thelen, 2003], ou encore, de permettre à ces aptitudes bas et hauts niveaux de partager un espace métrique commun [Tani, 2003]. Ainsi il est envisageable que le continu développement des capacités cognitives et motrices puisse amener à l'émergence de comportements aussi complexes que celui de l'utilisation d'outils [Lockman, 2000]. Et dans une perspective développementale, les comportements bas niveaux pourront former les composantes atomiques qui seront utilisées pour des apprentissages qualifiés de plus hauts niveaux [Stronger and Stone, 2006; Guerin, 2011].

Notons que Rolf et Asada ont quant à eux souligné le rôle crucial des buts [Rolf and Asada, 2015]. Ceux-ci sont souvent manuellement défini par la personne créant le modèle : or ils forment une question centrale pour la question du développement "open-ended", en particulier lorsque l'on considère les cas où ces buts ne pourront être connus en avance, et mettent ainsi en cause l'autonomie de développement d'un robot.

Nous pouvons résumer aux trois points suivants ce que requiert un système qui tendra à être "open-ended" d'un point de vue computationnel :

- Utiliser des fonctions de coût (ou de récompense) qui ne seront pas dédiées à une tâche particulière [Weng et al., 2001; Steels, 2004; Prince et al., 2005].

- Favoriser l'apprentissage de compétences réutilisables, qui pourront être directement applicables dans le cadre de scénarios non expérimentés au préalable (principe de la compositionnalité dans [Cangelosi et al., 2010], voir aussi [Oudeyer et al., 2007; Pezzulo and Castelfranchi, 2009]).
- Acquérir de nouvelles compétences tout en conservant celles acquises aux cours d'expériences passées [Schaal and Atkeson, 1998] (*i.e.*, ce que nous appellerons apprentissage incrémental).

Ces trois points nous semble ainsi devoir faire partie du cahier des charges qu'un modèle, ou qu'un ensemble de modèles interagissants les uns avec les autres, devrait idéalement satisfaire.

1.3 Conclusion

Dans la première partie de ce chapitre, nous avons étudié un ensemble de théories psychologiques ou neurobiologiques en parcourant les enjeux soulevés par notre problématique, et plus précisément en examinant différents problèmes que posent l'utilisation d'outils. Nous nous sommes donc intéressé à certaines questions fondamentales touchant au problème du lien entre moteurs et senseurs. Dans la suite de cette première partie, nous avons étendu notre état de l'art aux théories concernant la capacité à faire des séquences. Enfin, nous avons parcouru un ensemble d'autres mécanismes, plus transversaux, desquels nous nous inspirerons dans la suite de ce manuscrit.

Dans la seconde partie de ce chapitre, axée sur les problématique robotiques, nous avons commencé par questionner la question majeure de l'encodage sensorimoteur. Après avoir classé en deux principaux types distincts (qualifiés de "absolu" et "relatif") la manière de faire cet encodage, nous avons brièvement rendu compte des approches incontournables que sont le contrôle optimal, et l'apprentissage par renforcement, et nous avons évoqué certaines limitations qui y sont liées. Nous avons vu que l'approche PerAc permet de poser le problème différemment, en prenant comme facteur premier non un critère à optimiser en vue d'une tâche mais l'apprentissage du lien entre perception et action. Celui-ci, dans sa boucle avec l'environnement, permet un encodage sensorimoteur qui est en même temps une réponse à des contraintes environnementales. Il permet ainsi d'apprendre des réflexes comportementaux, dont l'adéquation avec l'environnement offre des solutions qui peuvent être perçues comme complexes, voire hauts niveaux du point de vue d'un observateur extérieur.

Ainsi cette thèse étudiera différents modèles sensorimoteurs afin d'en examiner les propriétés, et d'en proposer un nouveau à même de répondre à notre problématique, laquelle recouvre aussi bien le problème de l'encodage du schéma corporel, que celui de la capacité à faire des séquences.

Apprentissage de couplages sensorimoteurs

Dans cette partie, nous allons nous intéresser à différents modèles d’encodage sensorimoteur. Le but est d’explorer successivement différentes stratégies et d’observer le type de résultats et de propriétés que l’on obtient, notamment du point de vue de l’utilisation d’outils. Dans un premier temps (Sec. 2.1), nous allons nous baser sur de précédents travaux du laboratoire (voir [Rolland de Rengerve, 2013]), et explorer la stratégie appelée absolue, dans la mesure où celle-ci propose l’apprentissage du couplage entre les coordonnées dans l’espace sensoriel (la vision) et celles dans l’espace moteur (espace proprioceptif). Posant le constat que l’ajout d’un outil entraîne le ré-apprentissage complet d’un nouveau couplage sensorimoteur, nous proposerons dans un second temps (Sec. 2.2) un couplage permettant d’ajouter facilement une “transformation” (du type rotation ou translation) de ce couplage en nous basant sur le comportement des neurones dit à champ de gains (“gain-field”). Au couplage “absolu” est donc ajouté une transformation “relative”, permettant d’obtenir la plasticité d’encodage requise pour l’extension du schéma corporel liée à l’utilisation d’outils (voir Sec. 1.1.1). Mais ce sera alors l’architecture choisie par le modélisateur, et non l’exploration sensorimotrice elle-même, qui déterminera quels apprentissages seront absolus ou relatifs. Dans une dernière partie (Sec. 2.3), nous proposerons donc une première exploration d’un encodage totalement relatif, et nous expérimenterons quelques expériences exploitant les bénéfices d’un tel encodage.

2.1 Modèle “absolu” couplant des positions dans l’espace visuel et moteur

Le premier modèle auquel nous nous sommes intéressés propose l’encodage de neurones de lieu ainsi que le recrutement de transitions entre ces neurones (voir section 1.2.4). Ces neurones de lieu et le graphe de transitions entre ceux-ci peuvent être utilisés au sein d’une carte cognitive, laquelle permet notamment d’agir en vue d’un but, en pondérant les choix d’actions disponibles fournis par le graphe de transitions. Dans sa thèse, De Rengervé [Rolland de Rengerve, 2013] a proposé de porter le modèle des cartes cognitives au contrôle du bras de robot (voir Sec. 1.2.4). En utilisant un apprentissage sensorimoteur basé sur le maintien d’un homéostat, il a ainsi obtenu des comportements de type imitation immédiate, ou différée. En ajoutant l’utilisation d’une carte

cognitive, il a pu effectuer une tâche de tri de canettes dans laquelle, selon le contexte (traduit par la couleur de la canette) le bras katana dépose celles-ci dans l'emplacement correspondant. Dans des travaux communs, nous avons notamment travaillé sur les conséquences d'un signal de renforcement négatif lors d'une telle tâche [De Rengervé et al., 2012].

Dans cette section, nous regarderons dans quelle mesure cet apprentissage visio-moteur est utilisable pour l'utilisation d'outil. Nous commencerons par rappeler le modèle utilisé, et nous expliquerons les expériences qui ont été faites avec ce modèle et les conclusions que nous en avons tirées.

2.1.1 Apprentissage visuo-moteur

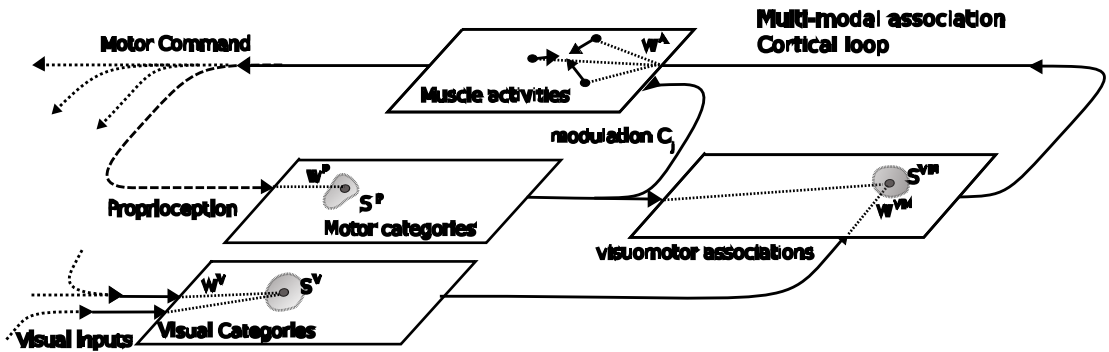


FIGURE 2.1 – Encodage sensorimoteur entre la vision et la proprioception. Un mécanisme de catégorisation (le SAW) encode la vision, la proprioception ainsi que l'association de ces deux catégories. Après apprentissage, lors d'une inadéquation entre ces deux modalités, des commandes motrices sont générées jusqu'à atteindre un état proprioceptif approprié. Issu de [Rolland de Rengerve, 2013].

Avec cette méthode, l'encodage sensorimoteur se fait en trois étapes, représentées sur la figure 2.1. Les entrées du modèle sont la proprioception (les positions articulaires du robot) et la vision (les positions x et y de la main). Chacune de ces modalités est catégorisée à l'aide d'un réseau de neurones, appelé "Selective Adaptive Winner" (SAW). Ce réseau utilise une règle proche de celle des K-moyens, et est couplé à un mécanisme de recrutement. Pour une entrée de dimension N , le calcul de l'activité (eq. 3.3) et de l'apprentissage (eq. 3.4) est donné par les équations suivantes :

$$A_k = 1 - \frac{1}{N} \sum_j |w_{kj} - e_j| \quad (2.1)$$

$$\Delta w_{kj} = \epsilon^{SAW} (e_j - w_{kj}) \quad (2.2)$$

$$\epsilon^{SAW} = \begin{cases} 1 & \text{si } A_k > \textit{vigilance} \\ 0 & \text{sinon} \end{cases} \quad (2.3)$$

où A_k est l'activité du $k^{ième}$ neurone de la couche de sortie du SAW, ϵ le taux d'apprentissage, w_{kj} le poids de la connexion entre le neurone de sortie k et le $j^{ième}$ neurone de la couche d'entrée dont l'activité est e_j . La variable *vigilance* est un seuil au-delà duquel le réseau effectue un recrutement. Au recrutement du neurone i , l'initialisation des poids se fait de la manière

suivante : $\forall j, w_{ij} = e_j$. Notons que le choix du seuil de vigilance détermine la granularité des états dans l'espace d'entrée.

Une troisième catégorie est apprise et associe les deux précédentes, celle motrice et celle visuelle. C'est ici que l'association sensorimotrice est apprise, et c'est l'homéostat qui devra être conservé, en envoyant les commandes motrices compensatoires. Ainsi, lorsque la catégorie motrice est en inadéquation avec la catégorie visuelle, du point de vue de cette association visio-motrice, alors une commande motrice est envoyée pour compenser l'écart constaté entre la catégorie motrice courante, et celle apprise lors de l'association visio-motrice, en se basant sur un modèle du muscle appelé Dynamic-Muscle PerAc [De Rengervé et al., 2015].

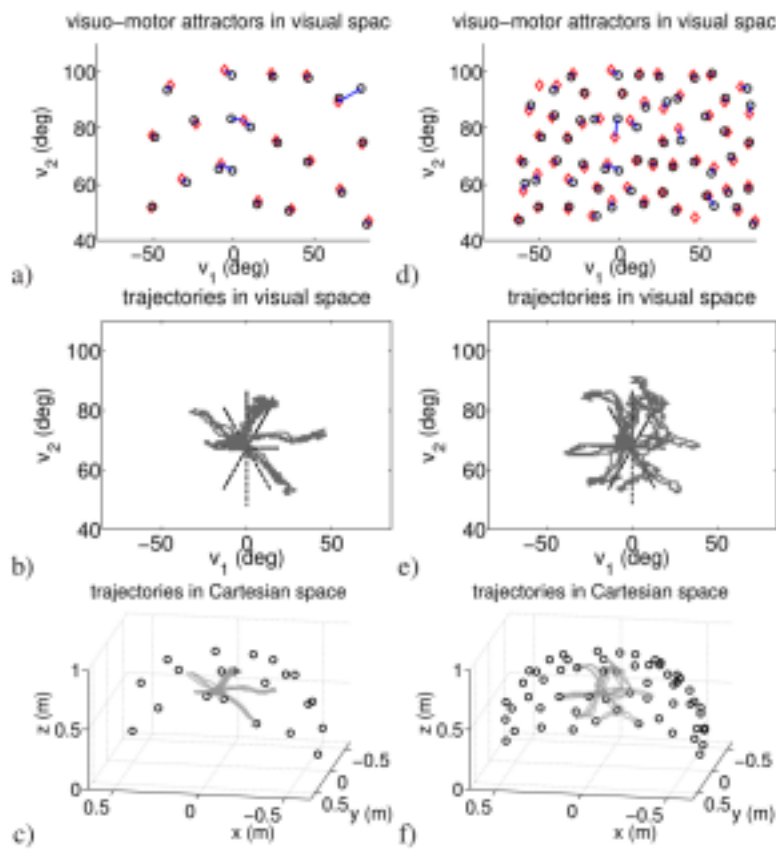


FIGURE 2.2 – Simulation de l'apprentissage en ligne d'attracteurs sensorimoteurs, avec 4 degrés de liberté et une caméra 2D. La colonne gauche représente les résultats obtenus avec une vigilance basse, et la colonne de droite avec une vigilance élevée. a) et d) représentent les états proprioceptifs (cercles noirs) et visuels (diamants rouges) recrutés. Chaque état proprioceptif est associé avec un état visuel (lien bleu). Après l'apprentissage, l'entrée visuelle forme artificiellement une étoile dans l'espace visuel (croix noire), et la reconnaissance visuelle entraîne un mouvement du bras vers l'endroit appris (ligne grise pointillée), ce que l'on observe dans b) et e). Dans c) et f) les mouvements résultants sont montrés dans l'espace cartésien 3D. Issu de [De Rengervé et al., 2015].

Considérant le développement frustré de la vision du bébé, dans ses premiers mois, il est possible d'envisager une ambiguïté de la perception qui conduit l'enfant à percevoir la main d'autrui comme étant la sienne. Une telle inadéquation entre la vision et la proprioception entraîne, de par

le mécanisme que nous avons décrit, lié au maintien d'un homéostat dans l'apprentissage visio-moteur, un comportement dit d'"imitation immédiate" dans lequel la main de l'enfant cherchera alors à rejoindre la main d'autrui observée [Rolland de Rengerve, 2013]. Les résultats de cette architecture, selon le degré de vigilance choisi, sont montrés dans la figure 2.2.

2.1.2 Application à l'utilisation d'outils

Afin de tester les possibilités offertes par ce modèle, nous avons expérimenté un tel encodage visio-moteur en ajoutant un outil au cours de l'expérience. Le but sera ici de conserver les propriétés de ce modèle provenant du maintien d'un homéostat, et les comportements offerts par l'ambiguïté de la perception. Les expériences qui suivent sont donc séparées en deux phases, celle de l'apprentissage visio-moteur, et celle de l'utilisation de l'homéostat issu de cet apprentissage dans des expériences dites d'imitation immédiate.

Nous basant sur les résultats de la psychologie et de la neurobiologie (voir sec. 1.1.1 et [Maravita and Iriki, 2004]), nous considérerons qu'un outil peut-être perçu comme une extension de soi-même. Du point de vue de notre modèle, cela signifie que l'entrée visuelle de l'outil pourra être assimilée à la main (c'est-à-dire à l'effecteur) du robot, que ce soit à la suite d'un apprentissage, ou encore de part une simple ambiguïté de la perception.

Expérience : ambiguïté de la perception avec un outil

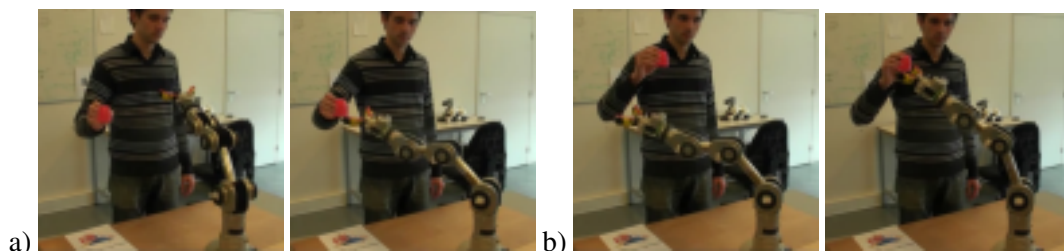


FIGURE 2.3 – Expérience 1, dite "d'imitation immédiate" dans laquelle l'homéostat, maintenu entre la vision de la main (ici la balle rose) et la proprioception associée du robot, est utilisé pour permettre à la main du robot d'atteindre la position supposée de la main du robot, qui se trouve être celle de l'expérimentateur tenant la balle rose. La main du robot suit donc celle de l'expérimentateur. Dans a), on voit dans la première image que la balle est éloignée (à dessein par l'expérimentateur) de la main robotique. Dans la seconde image, la main du robot a rejoint la position de la balle et se stabilise à cette position. b) montre un autre exemple d'une situation similaire.

En ce qui concerne l'ambiguïté de la perception, nous avons, dans une première série d'expériences, testé trois apprentissages différents. Chaque fois, ces apprentissages sont suivis du protocole d'imitation immédiate décrit précédemment (dans lequel l'entrée visuelle est substituée à une autre).

Dans le premier apprentissage visio-moteur (expérience 1), la vision de la main est associée à la proprioception de la main. En utilisant la position visuelle de la main (tenant un objet rose) de l'expérimentateur à la place de celle de la main du robot, l'expérience d'imitation immédiate fonctionne correctement (fig.2.3).

Dans un second apprentissage (expérience 2), nous associons à la proprioception de la main la position visuelle de l'outil (en forme de râteau, voir fig.2.4), lequel est saisi par la main.

Notons que dans cette configuration, la vision et la vigilance sont suffisamment précises pour que l'écart soit significatif.



FIGURE 2.4 – Outil, en forme de râteau, utilisé lors de nos expériences.

Puisque l'outil est dans la main, il y a une cohérence entre la position visuelle de cet outil, et la configuration proprioceptive au cours des déplacements. Ainsi, l'expérience d'imitation immédiate dans laquelle la position visuelle de la main (tenant toujours l'objet rose) remplace la position de l'outil ne pose pas plus de problème et donne des résultats satisfaisants (fig.2.5).

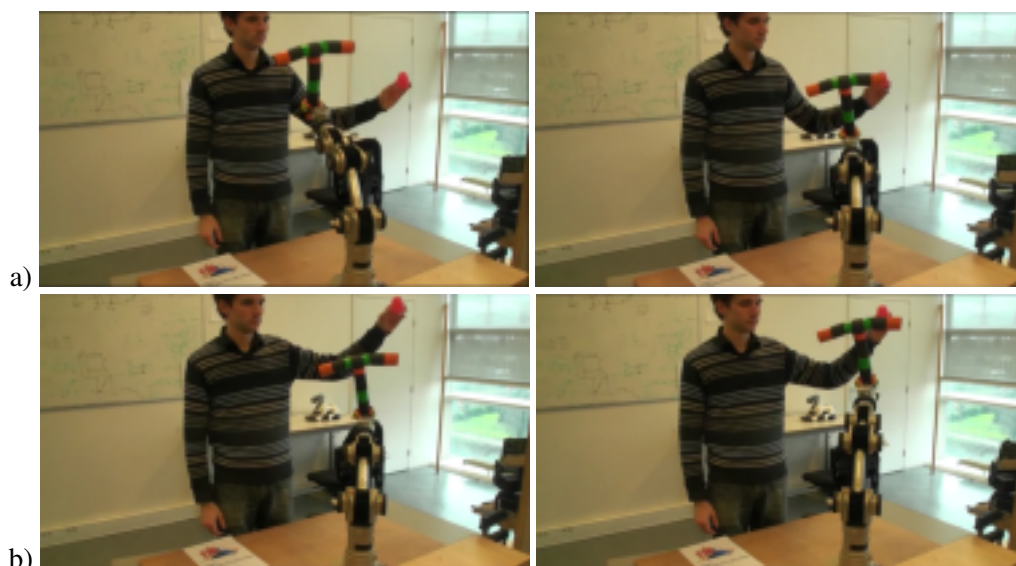


FIGURE 2.5 – Expérience 2 : même expérience que précédemment, mais l'homéostat appris repose cette fois sur la vision de l'outil, perçu comme correspondant à celle de la main (suivant l'hypothèse de l'extension du schéma corporel), associée à la proprioception. La main tenant l'outil suit donc la balle rose de l'expérimentateur selon le même mécanisme que précédemment. a) et b) illustre deux exemples dans lesquels l'outil, tenu par le main robotique, rejoint la position de la balle rose.

Le dernier apprentissage (expérience 3) est identique au premier, la configuration proprioceptive de la main est associée à l'entrée visuelle de la main réelle du robot. Toutefois, pour l'expérience d'imitation immédiate, nous avons donné comme pour la seconde expérience l'entrée visuelle correspondant à la position de l'outil, lequel est encore une fois situé dans la main. Cette expérience vise à reproduire les conditions de l'utilisation d'un outil non encore utilisé, après un premier apprentissage visio-moteur sans outil. Cette fois, le mécanisme précédemment expliqué engendre des commandes motrices qui vont faire se rapprocher la main du robot de ce qu'il perçoit comme étant sa main, à savoir l'outil. Or, puisque l'outil reste dans la main du robot

qui se déplace, celui-ci conserve sa distance initiale à la main : les commandes motrices visant à le rapprocher continuent alors indéfiniment jusqu'à atteindre une butée articulaire (fig.2.6).

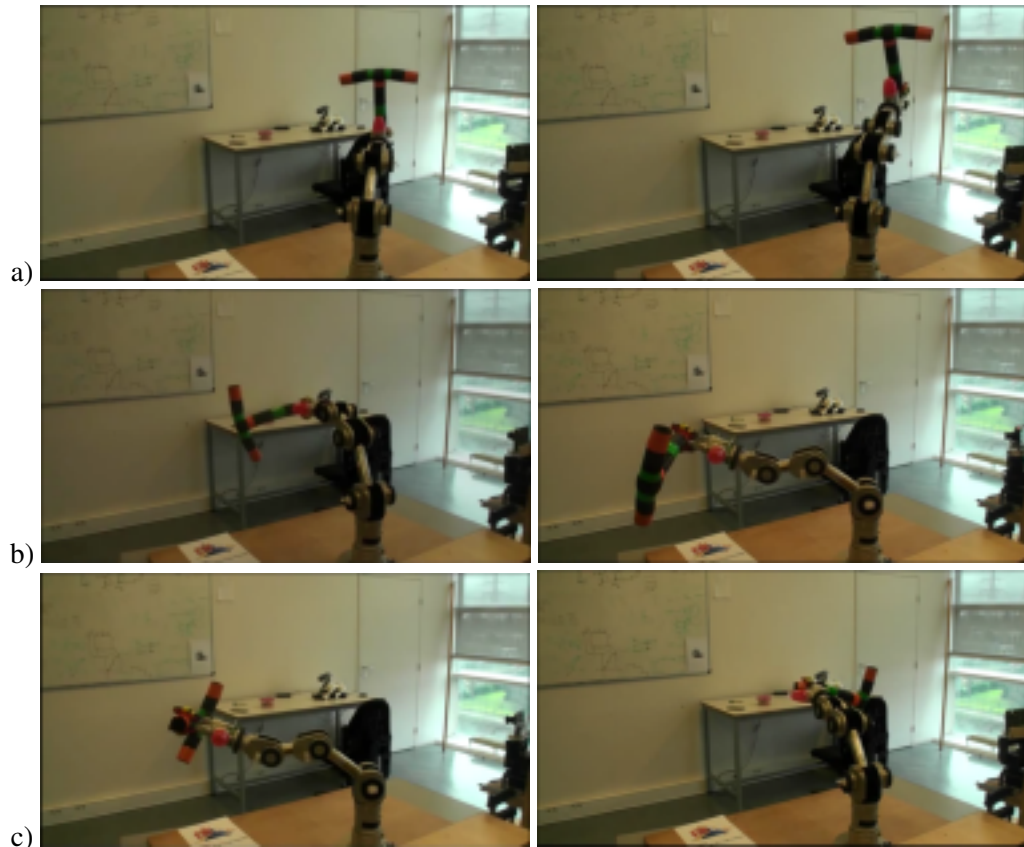


FIGURE 2.6 – Expérience 3 d’imitation immédiate, dans laquelle l’outil est perçu par le robot comme une extension de sa main mais après un apprentissage visio-moteur avec sa main réelle. Pour conserver l’homéostat, la main tente d’atteindre l’outil qu’elle tient, ce qui engendre un comportement de poursuite infinie. En a) l’outil est saisi de telle sorte qu’il est situé au dessus de la main. Le bras du robot se dirige donc vers le haut pour l’atteindre. Dans b), celui-ci est situé sur la gauche et cela engendre un mouvement infini vers la gauche, et en c), l’outil est à droite de la main, ce qui entraîne un mouvement de poursuite sur la droite.

Expérience : changement de la géométrie du bras

Les expériences effectuées ci-dessus ont donc été séparées en deux phases, celle de l’apprentissage visio-moteur, et celle de l’utilisation de l’homéostat issu de cet apprentissage, dans les expériences dites d’imitation immédiate. Dans cette partie, nous avons effectué une autre expérience (expérience 4), en reprenant l’expérience 3 mais ici nous avons laissé l’apprentissage se poursuivre après l’ajout de l’outil, afin d’observer le comportement du modèle.

Nous avons donc effectué un premier apprentissage permettant d’associer la proprioception du bras avec la vision de la main robotique. Nous avons obtenu les résultats de la figure 2.7 a). Les états visuels (représentés sous forme de diamants rouges) sont positionnés sur la figure en

se basant sur l'entrée visuelle de la main (reconnaissance de couleurs), et les états proprioceptifs (représentés sous forme de cercles noirs) sont obtenus en calculant les coordonnées de la main robotique dans l'espace cartésien à partir des valeurs angulaires de chaque articulation, en utilisant le modèle direct du bras.

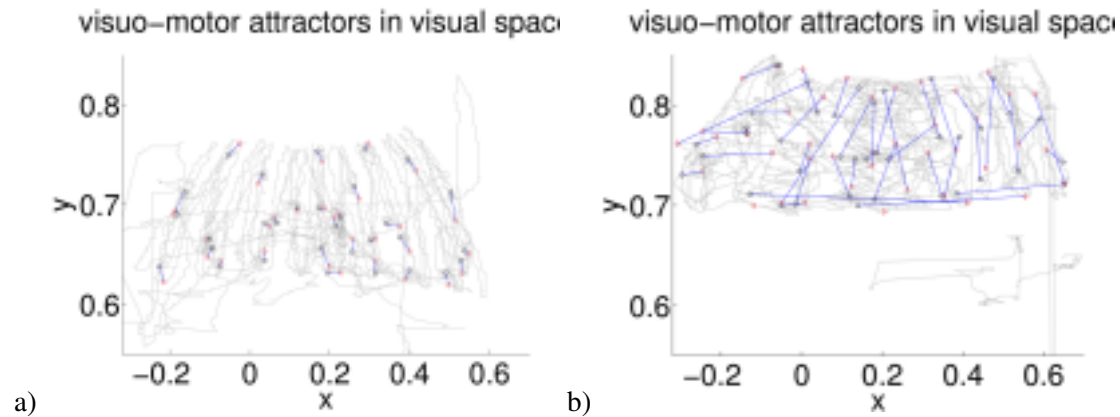


FIGURE 2.7 – Expérience 4 : apprentissage en ligne d’attracteurs sensorimoteurs, avec 4 degrés de liberté et une caméra 2D. Représente les états proprioceptifs (cercles noirs) et visuels (diamants rouges) recrutés. Chaque état proprioceptif est associé à un état visuel (lien bleu).

Les états visuels et proprioceptifs ne se superposent pas totalement à cause du bruit de la reconnaissance visuelle, mais les liens bleus, représentant l’association visio-motrice, sont malgré tout assez petits, ce qui marque la relative qualité de l’encodage du modèle dans ce cas.

Dans un second temps, nous avons ajouté l’outil afin d’observer les résultats obtenus après apprentissage, en permettant au robot de considérer cet outil comme étant sa main. Le robot prend alors les positions de l’outil comme entrée visuelle de son apprentissage visio-moteur. Dans cette expérience, l’outil (toujours le râteau de la fig.2.4) a été positionné vers le haut. La figure 2.7 b) montre ainsi que les états proprioceptifs et visuels ont tous été décalés vers le haut. On note que ce décalage vers le haut est important, d’une grandeur plus importante même que les distances entre états : ainsi, on ne pourrait baisser la valeur de vigilance jusqu’à ce que l’écart engendré par l’ajout de l’outil devienne imperceptible par le SAW, sans en même temps atteindre une granularité bien trop basse, qui effacerait du même coup presque toutes les distinctions entre les états.

Un réapprentissage est donc bien nécessaire, et on note cette fois de longs liens bleus, qui indiquent un décalage important entre la position visuelle de ce qui est perçu comme étant la main (biaisée par la présence de l’outil) et la position (théorique) de la main fournie par les données de la proprioception, en appliquant le modèle direct.

2.1.3 Conclusion

Nos expériences montrent que si ce modèle d’apprentissage visio-moteur, qualifié d’absolu, permet d’encoder efficacement les données sensorimotrices tout en permettant le comportement dit “d’imitation immédiate”, tant lorsque l’apprentissage concerne la main réelle (expérience 1) que l’outil (expérience 2), c’est au moment de l’ajout de l’outil qu’un problème se pose. En effet, il

est alors nécessaire (expérience 3) de réapprendre totalement les associations visio-motrices de la main (expérience 4) pour avoir un apprentissage utilisable.

Notons deux choses lorsque l'on retire l'outil de la main :

- Si le modèle n'apprend pas, alors il y aura un écart entre la position perçue de la main du robot et la proprioception. Et en voulant maintenir l'homéostat de l'apprentissage, la main se déplacera vers l'endroit appris, par exemple vers le bas si l'outil allait vers le haut : on observera un comportement similaire à celui de l'expérience 3 avec un mouvement permanent pour compenser un écart irréductible.
- Si le modèle apprend, alors un second réapprentissage, se ramenant à l'apprentissage initial qui aura été perdu, est à effectuer (expérience 4).

Il semble donc qu'un tel modèle ne soit pas le plus adapté pour une adaptation à la présence d'un outil dans la main, du moins lorsque celui-ci apporte une transformation à la géométrie du robot. Car alors le comportement lié à l'homéostat de l'apprentissage visio-moteur, utile pour atteindre des cibles grâce à l'ambiguïté de la perception, se révèle problématique dans ce cadre. L'apprentissage statique de positions absolues dans l'espace visuel ou proprioceptif, entraîne un réapprentissage total du robot lorsque sa géométrie varie brutalement (comme c'est le cas lors de l'ajout d'un outil).

Notons que, de par la capacité d'adaptation du SAW, un changement lent et progressif de la géométrie du robot (qui serait également à même de simuler, par exemple, la lente croissance de l'enfant), n'aurait pas le même effet. Les états concernés, en étant adaptés à chaque itération, déplaceraient ainsi leurs centroïdes (voir l'équation 3.4) ce qui leur permettraient de s'ajuster continuellement à ces évolutions (si du moins la vitesse d'apprentissage est "plus rapide" que ne l'est la croissance). Toutefois, cela ne correspond pas à de brusques changements de géométrie, comme ceux impliqués par l'utilisation ponctuelle d'outil.

Dans la partie suivante, nous nous sommes donc intéressés à la capacité de certains neurones, dit à champ de gains, à être le support de transformations sensorimotrices. L'idée est alors de voir dans quelle mesure il serait possible d'appliquer une transformation à un apprentissage pré-existant, plutôt que d'avoir à effectuer un ré-apprentissage complet.

2.2 Transformations sensorimotrices et neurones à champ de gains

Ce second modèle a pour but de s'inspirer de travaux de la neurobiologie afin de proposer une solution aux problèmes des transformations sensorimotrices, qui soit implémentable dans un robot.

Dans une première partie, nous verrons les travaux portant sur les neurones dit à champ de gains ("gain-fields" en anglais, noté GF), puis, dans une seconde partie, nous verrons le modèle que nous proposons et ses résultats (que l'on peut retrouver dans [Mahé et al., 2015]). Enfin, nous discuterons de ses avantages et de ses limitations.

Au cours des deux dernières décennies, les études du cortex post-pariétal (PPC) ont permis de mieux comprendre les mécanismes neuronaux impliqués dans la représentation spatiale du corps humain. Notre représentation corporelle est très plastique, et notre cerveau exploite pleinement l'ambiguïté perceptuelle liée à nos sens pour représenter spatialement non seulement nos membres, mais aussi les objets proches, et par extension les personnes autour de nous.

Dans cette perspective, les études d'Irîki, effectuées sur des macaques, sont particulièrement intéressantes car elles indiquent une réadaptation dynamique du schéma corporel lors de différentes situations [Irîki et al., 1996]. En jouant sur le retour visuel d'un téléviseur que le singe observe pour guider ses mouvements de bras, Irîki a montré comment les neurones pariétaux réajustent continuellement l'image corporelle (en l'occurrence celui de la main) de manière à être en adéquation avec de nouveaux cadres de référence [Okanoya et al., 2008]. Ceux-ci peuvent être issus de transformations spatiales complexes et non linéaires telles que la combinaison de translations, de changements d'échelle ou de rotations. Ce résultat a également été testé dans le cadre de l'utilisation d'outils, et les champs réceptifs spatiaux, associés à la main, des neurones pariétaux se sont étendus pour englober l'outil [Maravita and Irîki, 2004; Goldenberg and Irîki, 2007].

Les mécanismes neuronaux impliqués dans les représentations spatiales nécessitent simultanément des transformations non linéaires et un traitement rapide.

Le PPC est idéalement situé pour l'intégration multimodale, puisqu'il est l'une des premières structures corticales à recevoir les signaux sensoriels issus des différentes modalités [Andersen, 1997; Pouget and Snyder, 1997]. Cependant, il n'est pas simple de lier ces signaux, puisque chacun est encodé différemment et correspond à différentes parties du corps et à différents cadres de références. Ainsi, les personnes atteintes de lésions du cortex pariétal peuvent présenter des difficultés d'ajustement spatial, des troubles de la coordination ou encore des négligences spatiales [Keyzers, 2004].

La découverte de cellules associées aux mouvements d'atteintes, ou à des postures correspondant à une orientation particulière de la main, associée au contexte courant ou au plan moteur, a permis de mieux comprendre l'organisation fonctionnelle de ce type de réseau [Blohm and Crawford, 2009; McGuire and Sabes, 2009; Bremner and Andersen, 2012]. Andersen et collègues [Andersen and Mountcastle, 1983; Andersen et al., 1985] ont tout d'abord découvert des neurones, répondant à des commandes motrices spécifiques de saccades oculaires modulées par la position de l'oeil par rapport à la tête. Les résultats montrent que (1) ces neurones sont des neurones bimodaux, car ils codent deux informations à la fois, et (2) leur niveau d'amplitude est informatif, ce qui peut être modélisé. En outre, cet effet de modulation de gain, observé lors de ce type de comportement, ne correspond pas à celui d'une intégration nerveuse comme c'est le cas des neurones Intègre-et-Tire (en anglais "Integrate-and-Fire neurons"). Un modèle

mathématique plus adapté à cette réponse serait une intégration multiplicative entre les signaux sensoriels entrants, ce qui peut être approximée en une fonction de base non linéaire [Pouget and Snyder, 1997].

Dans de précédents travaux, il a été montré comment le mécanisme de modulation de neurones GF peut être, à la manière de certains neurones pariétaux, appliqué à l'intégration de signaux audio-visuels et proprioceptifs dans un robot comprenant une tête, un cou et des yeux [Pitti et al., 2012]. Les neurones GF ont été utilisés pour relocaliser l'emplacement d'un signal sonore (dans un cadre de référence centré sur la tête) en coordonnées de la rétine (dans un cadre de référence centré sur les yeux). Le modèle de GF a permis au système d'améliorer la précision avec laquelle est estimée la position d'un stimulus visuel (en l'occurrence la position de la bouche quand une personne parle), en utilisant le signal sonore qui y est associé.

Dans nos expériences, un robot constitué d'un bras et d'une caméra va tout d'abord effectuer un apprentissage des associations visuomotrices à l'aide de GF. Puis, une fois son schéma corporel appris, un second module de GF est ajouté pour apprendre à accéder à un nouvel "ensemble de tâches" sensorimotrices (en anglais, "task-set") [Pitti et al., 2013a, 2009], lequel doit être appris notamment suite à des transformations liées à des mouvements de la tête (rotations, translations), ou à l'utilisation d'outils.

2.2.1 Modèle

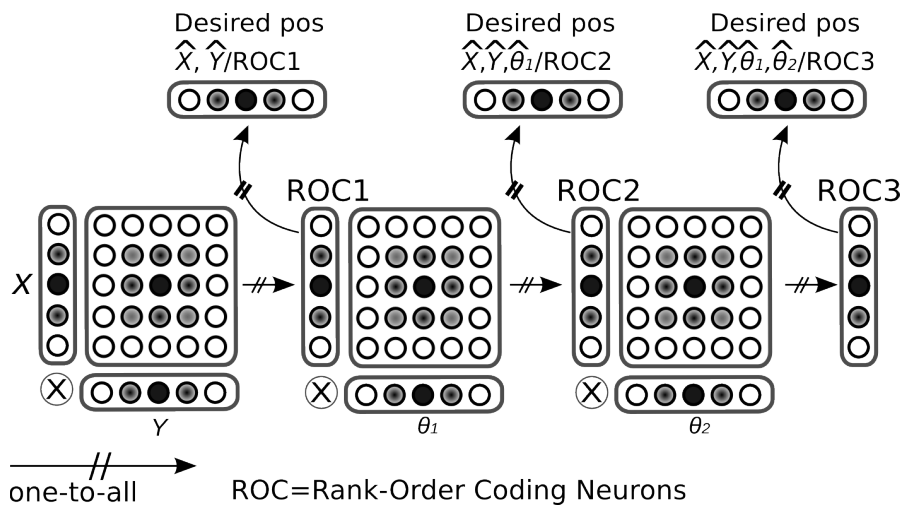


FIGURE 2.8 – Architecture neuronale pour neurones GF. Ceux-ci lient deux activités, et en réutilisant les neurones GF résultant de la multiplication de deux modalités comme l'une des deux modalités d'une autre carte de GF, ils peuvent ainsi lier l'activité de plusieurs entrées. Ces neurones peuvent servir de fonctions de base à partir desquelles les représentations souhaitées peuvent être cartographiées. (ROC) représente l'algorithme de "Rank-Order Coding". Les signaux sensoriels I peuvent alors être estimés à partir de la combinaison linéaire des neurones ROC. La sortie souhaitée \hat{I}/ROC , est alors l'estimation du signal I par ROC.

Notre architecture implémente des neurones multiplicatifs, appelés neurones GF, qui multiplie, unité par unité, la valeur de deux populations neurales entrantes, ou plus voir fig. 2.8. Son fonctionnement est similaire à celui des fonctions de base radiale (RBF), car les GF trans-

forment les signaux d'entrées en un ensemble de fonctions de base, un espace fonctionnel, qui peut également être exploité pour représenter les stimuli dans différents référentiels [Salinas and Thier, 2000; Pouget and Snyder, 1997]. La multiplication entre les signaux sensoriels afférents de deux modalités différentes (M_1, M_2) est la multiplication par éléments de deux distributions de probabilité X_{m_1} et X_{m_2} , deux vecteurs de dimensions $\{\mathbb{M}_1$ et $\mathbb{M}_2\}$ respectivement, avec $\{m_1, m_2 \in \mathbb{M}_1, \mathbb{M}_2\}$. Nous considérerons la matrice dont l'activité $X_m, m \in \mathbb{M}_1 \times \mathbb{M}_2$ est apprise par les neurones GF. L'équation du n -ième neurone GF X_n^{GF} , $n \in N$, dont le poids synaptique $w_i, i \in \mathbb{M}_1, \mathbb{M}_2$ est donnée par :

$$X_m = X_{m_1} \times X_{m_2} \quad (2.4)$$

$$X_n^{GF} = \sum_{i=0}^{\mathbb{M}_1 \times \mathbb{M}_2} w_i X_i \quad (2.5)$$

L'idée ici est que la carte des X^{GF} encode une combinaison de deux valeurs, et l'amplitude du GF encode une modalité de manière conditionnelle, et dans une dimension de plus petite taille [Braun et al., 2009, 2010]. Cette caractéristique est utilisée pour modéliser différents signaux sensoriels en tant que combinaison linéaire f de neurones GF, ainsi $f(X_n^{GF}) = \sum_{n=0}^N \omega_n X_n^{GF}$, avec $n \in N$ neurones GF, et ω la pondération.

Ces neurones apprennent les associations entre diverses modalités de la manière suivante (voir fig. 2.8) : une fois une première association effectuée, disons entre les variables X et Y de la fig. 2.8, il est possible de chaîner les cartes de neurones de telle sorte que l'activité des dernières (par exemple, codant la nouvelle modalité θ_1) dépend de l'activité des premières (en l'occurrence des modalités X et Y). A titre d'exemple, voici les équations pour la deuxième carte X_n^{GF2} , construite à partir de l'activité de la première carte $f(X_n^{GF1})$ et de la nouvelle modalité X_{m_3} , un vecteur de dimension \mathbb{M}_3 :

$$X'_m = X_{m_3} \times f(X_n^{GF1}) \quad (2.6)$$

$$X_n^{GF2} = \sum_{i=0}^{\mathbb{M}_3 \times N} w_i X'_i \quad (2.7)$$

Les nouveaux neurones GF X_n^{GF2} dépendent de X_{m_3} et $f(X_n^{GF1})$ et encodent alors une carte relative à l'activité de toutes les modalités précédentes, X_{M_1} et X_{m_2} . Notons que X^{GF1} représente la multiplication des valeurs des deux modalités d'entrée, et donc ne représente pas de manière unique la combinaison de ces deux valeurs. Cependant, d'un point de vue biologique, l'observation d'un seul neurone GF montre un comportement similaire, au sein des neurones pariétaux [Salinas and Thier, 2000]. Alors qu'une seule carte de GF n'est pas suffisante pour résoudre l'ambiguïté de certains cas, la combinaison linéaire de multiples cartes (X^{GF1} et X_n^{GF2} par exemple), peut aider à la généralisation. Ainsi, les neurones GF représentent chacun un modèle d'activité global spécifique de la carte X^{GF} , qui code une combinaison de paires de valeurs. Ce faisant, le système retranscrit l'information d'une modalité relativement aux autres. Ce processus multimodal est semblable au chaînage de règles conditionnelles bayésiennes entre plusieurs variables [Deneve and Pouget, 2004]. Comme le nombre de dimensions augmente, une combinaison linéaire d'un nombre approprié de neurones GF peut permettre de représenter les données d'entrée "suffisamment bien", si l'espace d'entrée est suffisamment faible. Dans un tel cas, nous ne devrions pas observer de perte de résolution après la phase d'apprentissage.

L'algorithme de "Rank-Order Coding"

Dans la continuité de précédent travaux [Pitti et al., 2012, 2013b], nous avons utilisé une règle proposée par Thorpe et collègues [Van Rullen et al., 1998; Thorpe et al., 2001; Van Rullen and Thorpe, 2002] appelé algorithme "Rank-Order Coding" (ROC).

La sortie des neurones X^{GF} est calculée en multipliant non pas directement les valeurs d'amplitude du vecteur de signaux sensoriels I , mais l'inverse de son rang $rank(I)$ par les poids synaptiques w ; $W \in [0, 1]$.

Pour un vecteur d'entrée de dimension M et pour une population de N neurones GF X , nous remplaçons l'eq. 2.5 by :

$$X_n^{GF} = \sum_{m \in M} \frac{1}{rank(I_m)} w_{n,m}^{GF} \quad (2.8)$$

La règle de mise à jour des poids des neurones est similaire à l'algorithme d'apprentissage "Winner Takes All" des cartes auto-organisées de Kohonen [Kohonen, 1982]. Pour le neurone gagnant $s \in N$ et pour tous les signaux afférents $m \in M$, nous avons :

$$\Delta w_{s,m}^{GF} = \alpha \left(\frac{1}{rank(I_m)} - w_{s,m}^{GF} \right), \quad (2.9)$$

Où α est le taux d'apprentissage. Nous avons choisi $\alpha = 0.01$. Notons que de manière similaire aux RBF, plus le signal d'entrée est distant au neurone, plus son niveau d'activité est bas, ce qui permet de les utiliser comme champs réceptifs.

2.2.2 Apprentissage de transformations sensorimotrices

Une fois que les neurones GF ont appris les règles visuomotrices à partir de leur espace d'entrée, ils peuvent être exploités pour comparer toutes les inadéquations visuomotrices constatées, lors de changements de perception physique ou encore d'ambiguïtés sociales. Le modèle de comparaison, représenté sur la figure 2.9, calcule la différence entre la valeur d'un signal réel A avec une valeur estimée \hat{A} en sortie de notre réseau de GF. Ceci peut alors, par la suite, servir à produire des transformations rendant compte des erreurs constatées.

En ce sens, les neurones GF utilisés dans notre modèle sont proches des fonctions de base radiales (RBF) utilisées dans le traitement d'image dans lesquelles la combinaison linéaire de RBF peut servir à déformer l'espace d'entrée vers un modèle particulier.

Considérons le point A et le nouveau point \hat{A} estimé par le réseau de GF, avec :

$$\begin{aligned} f(A) &= \sum_{i=1}^n w_i X_i^{GF}(A) \\ f(\hat{A}) &= \sum_{i=1}^n w_i X_i^{GF}(\hat{A}) \end{aligned} \quad (2.10)$$

Exprimons $f(\hat{A})$ en fonction de $f(A)$:

$$\begin{aligned} f(\hat{A}) &= \sum_{i=1}^n w_i (X_i^{GF}(A) + (X_i^{GF}(\hat{A}) - X_i^{GF}(A))) \\ f(\hat{A}) &= \sum_{i=1}^n w_i X_i^{GF}(A) + \sum_{i=1}^n w_i (X_i^{GF}(\hat{A}) - X_i^{GF}(A)) \\ f(\hat{A}) &= f(A) + \sum_{i=1}^n w_i (X_i^{GF}(\hat{A}) - X_i^{GF}(A)) \end{aligned} \quad (2.11)$$

Cela signifie que plus \hat{A} sera proche de A , plus $f(\hat{A})$ sera proche de $f(A)$, autrement dit plus un point \hat{A} sera proche d'un point A utilisé pour l'entraînement des neurones GF, plus la transformation associée sera proche de la transformation donnée en exemple.

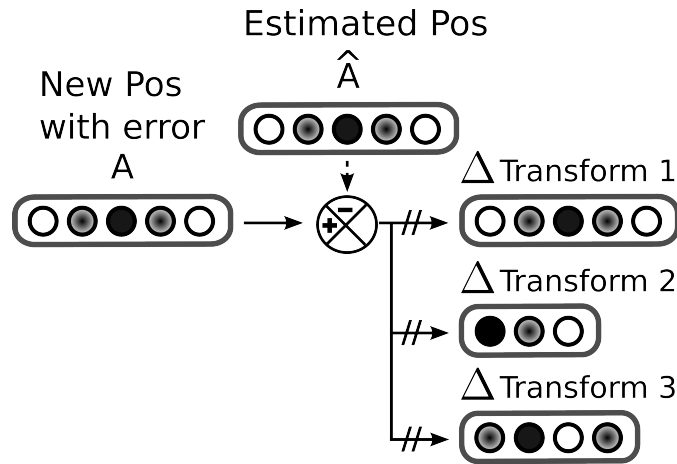


FIGURE 2.9 – Mécanisme de transformation avec signal d’erreur. On peut calculer un signal d’erreur ΔA entre une variable A et son estimation \hat{A} effectuée par le réseau neuronal de la figure 2.8. Cette erreur peut être apprise par différentes cartes en fonction du contexte auquel il correspond. La carte d’erreur peut alors servir à apprendre la transformation permettant de passer du point A au point \hat{A} .

Le second terme correspond à la fonction de transformation qui minimise la distance entre $X_i^{GF}(A)$ et $X_i^{GF}(\hat{A})$. Un perceptron peut estimer cette fonction autour du point A en identifiant les poids qui minimisent le problème de correspondance.

2.2.3 Expériences

Dans nos expériences, nous utilisons le bras robotique Kinova avec 7 degrés de liberté et une caméra fixe pour l’observer, comme le montre la figure 2.10 a).

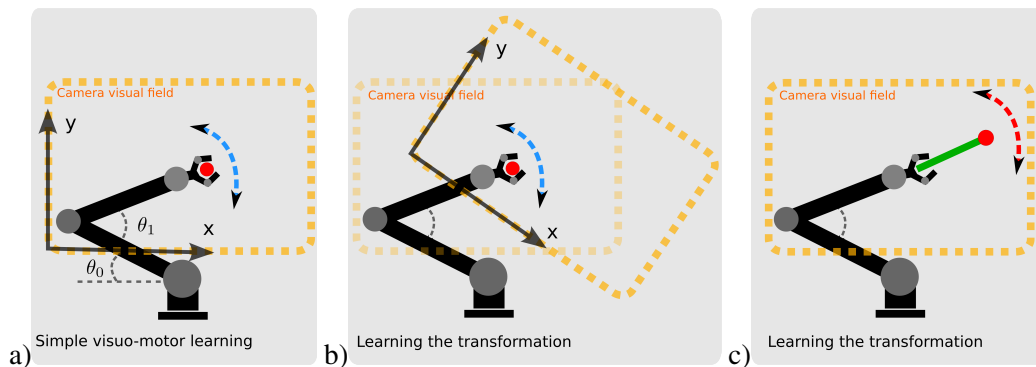


FIGURE 2.10 – Cadre expérimental pour l’apprentissage de transformations sensorimotrices. (a) Cette première situation correspond à l’apprentissage de la correspondance visuomotrice entre les variables motrices (deux degrés de liberté) et la position (X, Y) dans le champ visuel de la caméra. Bien que cette relation soit non linéaire, une légère modification du référentiel ne nécessite pas de réapprendre tous les liens visuomoteurs mais plutôt la fonction de transformation responsable de l’erreur globale. Ces changements peuvent être dus à une transformation visuelle, comme c’est le cas pour une rotation de caméra (b), à un changement corporel comme c’est le cas lors de l’utilisation d’outil (c).

Par souci de simplicité, nous avons limité le bras à deux degrés de liberté (DoF) dans le plan visuel de la caméra, mais l'architecture peut être facilement étendue à plus DoFs en chaînant les cartes GF. Les deux articulations motrices correspondent respectivement aux articulations de l'épaule et du coude, θ_0 et θ_1 , avec $\theta_0 \in [0^\circ, 200^\circ]$ et $\theta_1 \in [0^\circ, 100^\circ]$.

Chaque angle moteur est codé en un vecteur discrétisé de 22 valeurs suivant une gaussienne centrée sur l'angle moteur actuel. Un système de vision basé sur la couleur fournit les coordonnées (x, y) de la couleur rouge, correspondant à la main, dans le cadre de référence de la rétine. De la même manière, avec les angles moteurs, les coordonnées visuelles sont codées en deux vecteurs discrétisés de 22 valeurs suivant une gaussienne centrée sur la position courante. En utilisant l'architecture de GF de la figure 2.8, le système est composé de trois cartes où chaque groupe ROC est composé de 22 neurones. Ainsi, le dernier groupe ROC apprend à associer les angles moteurs (θ_0, θ_1) aux coordonnées de la main selon les axes (x, y) . La sortie du réseau GF donne alors une représentation, basée sur une RBF, des coordonnées apprises de la main.

Pour tirer profit des GF, un perceptron sur la couche de sortie adaptera sa sortie aux coordonnées estimées de la main dans le champ visuel $((\hat{x}, \hat{y}))$ et dans les angles moteurs $((\hat{\theta}_0, \hat{\theta}_1))$. Cela correspond à l'intégration des quatre entrées à travers les trois cartes chaînées (voir fig. 2.8).

Nous utilisons donc cette couche de sortie pour apprendre les transformations produites dans des situations d'ambiguïté perceptuelles, comme le déplacement de la caméra, ou l'utilisation d'outils (voir fig. 2.10 (b) et (c)). Enfin, une dernière couche est ajoutée pour apprendre la différence entre les valeurs réelles et estimées des coordonnées de la main après qu'une transformation ait été appliquée au système (voir eq. 2.11). La sortie de cette couche de neurones fournit alors une transformation qui donne, pour tout point donné, la transformation censée annuler celle réellement appliquée au système.

2.2.4 Résultats

Le robot apprend sa coordination visuomotrice à l'aide des neurones GF. Nous vérifions ensuite comment s'est effectué cet apprentissage, en observant les coordonnées spatiales de la main, tant dans l'espace visuel que dans l'espace peripersonnel.

Tout d'abord, le robot effectue un babillage moteur de façon aléatoire afin que les neurones ROC remplissent uniformément l'espace visuomoteur. Après cette période d'apprentissage, qui a durée 12 minutes et correspond à 12000 échantillons, correspondant à une nouvelle configuration de bras toutes les 0,05 secondes, les neurones ROC s'auto-organisent eux-mêmes et reproduisent le comportement des neurones GF du PPC.

Pour illustrer cette phase, nous avons tracé dans la figure 2.11 le niveau d'activité de trois différents neurones ROC de la troisième carte, par rapport aux paires de variables (θ_0, θ_1) et (x, y) . Le code de couleur indique la valeur d'activité pour chaque neurone et leurs champs réceptifs respectifs dans le domaine moteur et dans le domaine visuel.

Notons que les champs récepteurs visuels et moteurs ne sont pas séparés les uns des autres, et le niveau d'activité des neurones GF est lié aux quatre variables. Dans la figure 2.12 par exemple, le niveau d'amplitude d'un neurone GF varie par rapport à l'angle moteur θ_0 pour une plage donnée de valeurs sur l'axe X. Le code couleur indique trois intervalles différents de l'angle moteur θ_0 . On voit que son champ réceptif n'est pas strictement centré sur une valeur particulière de l'axe X, mais se déplace par rapport au retour proprioceptif, phénomène également observé dans les neurones biologiques du PPC [Salinas and Thier, 2000].

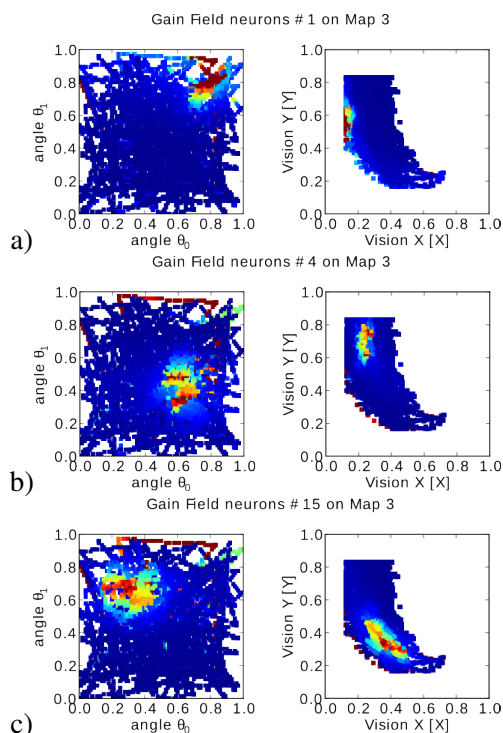


FIGURE 2.11 – Activité des neurones GF. Les neurones 1, 4 et 15 resp. a), b) et c) encodent différents emplacements sensorimoteurs dans (θ_0, θ_1) et (x, y) . Chaque variable est normalisée dans l’intervalle $[0, 1]$.

Pour estimer les propriétés de notre système à généraliser les transformations visuomotrices, nous avons effectués des expériences sur trois transformations visuelles différentes, en s’inspirant des travaux de Iriki, qui a montré sur un macaque une adaptation rapide d’une nouvelle position visuelle de ses membres lors de changements opérés sur un écran [Iriki et al., 1996] ou sur le corps [Maravita and Iriki, 2004; Okanoya et al., 2008]. La première transformation consiste en une légère rotation de la caméra dans le plan du bras robotique. La seconde correspond à une extension de la main du robot avec un outil, et la troisième consiste en une translation de la caméra, en suivant le plan des mouvements du bras (voir fig. 2.10 (b) et (c)).

Dans la figure 2.13, nous présentons les transformations estimées correspondant à trois cas a), b) et c), respectivement celui de la rotation, de l’utilisation d’outils et de la translation.

Chaque neurone du perceptron de la couche de sortie estime les valeurs $(\Delta x, \Delta y)$ apprises à partir des valeurs réelles (x, y) , ainsi que de celles attendues (x', y') provenant des cartes de GF.

Afin de visualiser les transformations qui relient les deux espaces sensorimoteurs, nous avons tracé des flèches liant chaque point (x, y) avec le point après transformation qui y est associé $(x + \Delta x, y + \Delta y)$. Dans les figures 2.13, on voit en bleu la trajectoire de la main avant transformation, et en rouge la trajectoire de la main après transformation. Notons que la longueur des flèches ne correspond pas à l’amplitude réelle de la transformation, mais est réduite pour qu’elles soient mieux visualisées sur la figure. La figure 2.13 a) affiche la transformation correspondant à la rotation de la caméra. Comme on peut le voir, les flèches au centre, là où

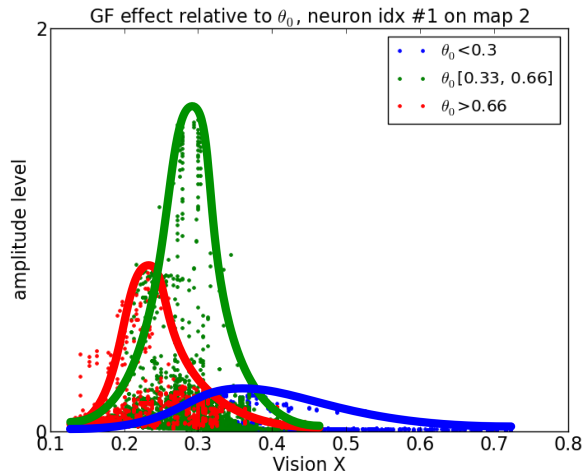


FIGURE 2.12 – Dynamique de l’amplitude d’un neurone GF par rapport à la variable motrice θ_0 sur l’axe X . Notons que les deux variables sont liées ensemble de par leur amplitude. Les centres sur X se déplacent légèrement aussi par rapport à la variable θ_0 comme on le voit également dans les neurones biologiques.

se déplace le robot, approximativement correctement la rotation. Ainsi la généralisation effectuée sur un voisinage proche des exemples appris est relativement bonne. Mais au-delà d’une certaine distance par rapport aux exemples appris (soit, dans notre cas, à l’extérieur de la zone centrale), nous pouvons voir des estimations peu précises, et parfois mauvaises, dans lesquelles l’orientation des flèches ne correspond plus vraiment à la direction de la rotation.

La deuxième transformation de la figure 2.13 b) correspond à l’ajout d’un petit bâton à l’extrémité du bras, qui étend celui-ci vers le haut. Au cours de cette expérience, le système de vision interprète la position de l’extrémité du bâton comme étant celle de la main. Dans ce cas également, la transformation est correctement apprise, et cette fois la généralisation est efficace dans une partie de l’espace des paramètres plus important, ce qui est très certainement dû à la simplicité de cette transformation.

La dernière transformation effectuée est un simple décalage vertical de la caméra (voir fig. 2.13 c). Une fois de plus, la transformation est correctement apprise, et cette fois la généralisation s’applique efficacement à presque tout l’espace des paramètres (ce qui s’explique par le fait que la transformation apprise est véritablement globale).

D’autres résultats, notamment sur le problème dit de la correspondance lors d’interactions interpersonnelles, sont à trouver dans l’article que nous avons écrit sur ce sujet [Mahé et al., 2015].

2.2.5 Conclusion

Dans cette section, nous avons présenté une architecture basée sur les neurones GF susceptible d’encoder l’information sensorimotrice en utilisant des propriétés multiplicatives. Le chaînage de ce type d’opérations multiplicatives, présenté à la figure 2.8, permet d’avoir en sortie des neurones GF encodant bien plus de deux informations sensorielles, en utilisant des multiplications du type $X_m = X_{m_1} \times X_{m_2}$ (voir eq.2.4), permettant de calculer une valeur du neurone

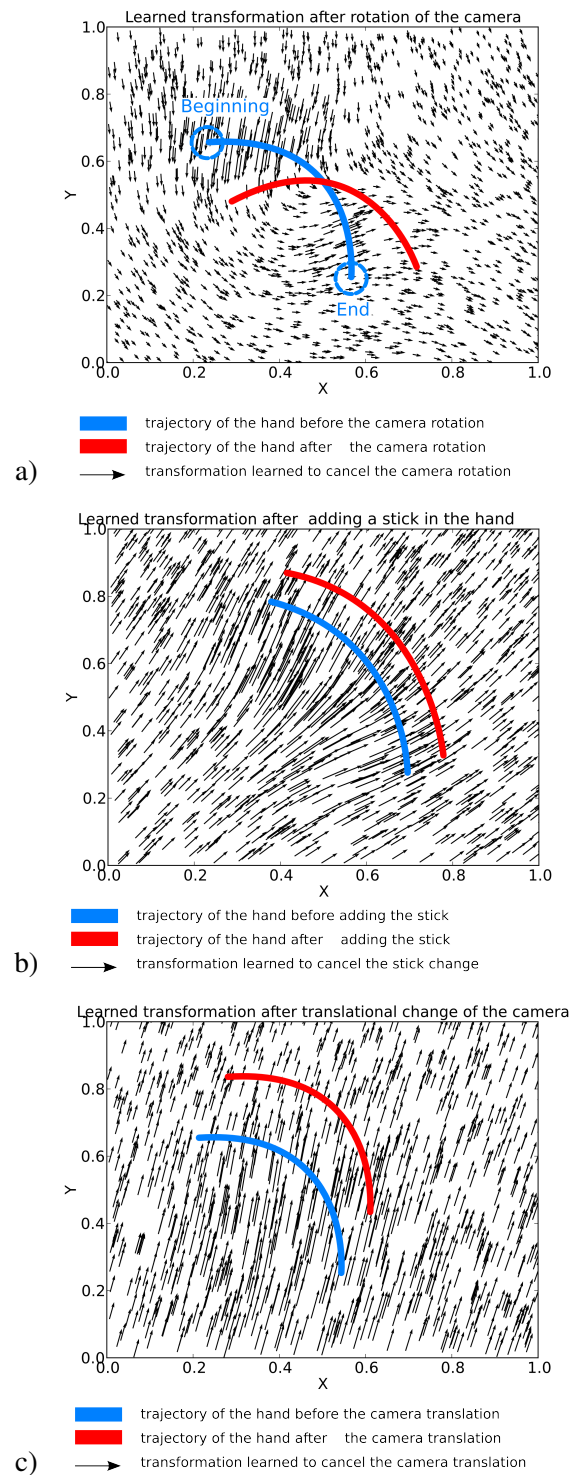


FIGURE 2.13 – Transformations par rotation, utilisation d’outils et translation. Les flèches correspondent aux transformations locales estimées, les lignes bleues correspondent aux trajectoires originales de la main du robot avant transformation, et les lignes rouges aux trajectoires après transformation. (a) Transformation apprise après rotation de la caméra. (b) Transformation apprise après avoir ajouté un bâton à la main robot. (c) Transformation apprise après translation de la caméra.

GF X_n^{GF} , suivie du type $X_m' = X_{m_3} \times f(X_n^{GF})$ (voir eq.2.6), et ainsi de suite. Outre la ressemblance de ce type de neurones avec ceux que l'on retrouve dans le PPC, ceci permet un encodage de la sensorimotricité qui permet de facilement ajouter de nouvelles opérations les unes à la suite des autres, et ainsi de conserver l'apprentissage précédent tout en ajoutant un correctif en cas de constat d'erreur. Notons que l'idée d'ajouter un correctif permettant de ne pas perturber l'apprentissage précédent sera reprise et développée dans le chapitre 4.

Toutefois, le système actuel présente deux limites. La première est que le chaînage est choisi par le modélisateur, et non déterminé par la sensorimotricité du robot. Or l'ordre des opérations a une influence sur le résultat obtenu. De plus, le fait qu'il soit statique empêche d'ajouter facilement de nouvelles modalités, ou de nouvelles transformations à la volée. Si toutes les modalités ne sont pas pertinentes à ajouter a priori pour le modélisateur (comme le tactile dans notre expérience), notons que cette information peut s'avérer utile pour reconnaître un contexte lié à l'usage d'un outil.

Ce qui nous amène à la seconde limitation. Celle-ci concerne l'absence de moyen de lier l'erreur constatée, et potentiellement corrigée, à un contexte défini. En effet, notre architecture permet d'apprendre à corriger une erreur en apprenant une transformation, laquelle peut correspondre à une translation, une rotation, ou encore à l'utilisation d'outil [Mahé et al., 2015]. Si un contexte particulier rend cette transformation nécessaire, il manque un moyen d'attribuer une transformation donnée à un contexte qu'il resterait à déterminer. Ceci permettrait en effet de profiter du fait que le correctif appris ne perturbe pas le précédent apprentissage, et peut donc être dynamiquement activé ou désactivé en fonction de la reconnaissance du contexte courant.

Notons que l'idée que nous venons de présenter est une alternative à une autre méthode qui consiste à déterminer la transformation (de manière plus générale, l'apprentissage pertinent) à partir du signal d'erreur lui-même (voir par exemple la notion de responsabilité dans [Wolpert et al., 2003]). Au lieu de déduire le contexte, nous proposons de l'apprendre et de prédire les transformations associées aux différents contextes appris. Comme nous le verrons dans le chapitre suivant (chapitre 3), l'inversion de cet apprentissage (dans un sens similaire à celui du modèle inverse) permettrait également de rendre de tels contextes "désirables".

De fait, la question centrale à laquelle nous proposerons d'apporter des solutions sera celle de l'apprentissage de covariations : nous venons d'évoquer que la loi sensorimotrice varie quand le contexte varie et cette idée sera développée dans les chapitres suivants. Mais dans la section qui suit nous allons nous intéresser à la simple covariations de senseurs et de moteurs, et aux propriétés émergentes d'un tel encodage sensorimoteur.

Nous venons d'évoquer que la loi sensorimotrice varie quand le contexte varie, et cette idée sera développée dans les chapitres suivants. Par ailleurs notons que, de fait, la question centrale à laquelle nous proposerons d'apporter des solutions à partir d'ici sera celle de l'apprentissage de covariations. Dans la section qui suit nous allons nous intéresser à la simple covariations de senseurs et de moteurs, et aux propriétés émergentes d'un tel encodage sensorimoteur.

2.3 Modèle couplant flot optique et mouvements oscillatoires

Contrairement aux parties précédentes, dans cette partie nous nous sommes intéressés à un modèle apprenant non pas des associations entre des points dans un espace statique, mais apprenant des associations entre mouvements provoqués par les mouvements du robot, et mouvements perçus. Afin d’explorer les possibilités offertes par de telles associations, nous nous sommes intéressés à un modèle couplant entre elles des primitives bas niveaux, visuelles et motrices, basées sur le mouvement. Pour les primitives motrices, nous avons utilisé de simples oscillateurs actionnant certaines synergies motrices. Pour les primitives visuelles, nous avons utilisé une estimation du flot optique perçu. Grâce à l’apprentissage d’association entre les deux, et en ajoutant un modèle permettant la reconnaissance de formes basé sur la reconnaissance des vues locales en mouvement, nous avons, dans une approche développementale, effectué des expériences permettant d’obtenir un comportement d’atteinte (de “reaching”) bas niveau.

Dans une première partie, nous allons décrire les primitives visuelles et motrices implémentés pour nos expériences. Nous verrons ensuite comment nous avons proposé d’apprendre ensemble ces primitives, afin d’obtenir un comportement proche de l’imitation immédiate décrite en Sec. 2.1.1. Puis, nous décrirons le système visuel permettant de reconnaître la forme du bras, qui sera apprise durant la phase de babillage moteur. Enfin, nous verrons dans une dernière partie comment tirer profit d’un tel apprentissage relatif couplé à une reconnaissance de formes, pour faire émerger, en jouant de nouveau sur l’ambiguïté de la perception, un comportement de “reaching”.

2.3.1 Primitives visuelles et motrices

Primitives motrices basées sur des oscillateurs

De récentes études ont montré que la réponse du cortex moteur durant un comportement de “reaching” contenait une composante oscillatoire, courte mais marquée, même lorsque le mouvement lui-même ne contient pas d’oscillations [Churchland et al., 2012]. En s’inspirant de ces études neurobiologiques, nous avons exploité la notion de configuration rythmique, et proposons d’utiliser un contrôle moteur basé sur des oscillations. Ce choix a également été fait pour éviter d’utiliser un contrôleur moteur doté d’une précision que l’on ne peut attendre dans les premiers stades du développement.

Pour ces expériences, nous avons de nouveau utilisé le bras Katana. Le contrôleur moteur est illustré figure 2.16-A. Chaque articulation du bras Katana est contrôlée par un ensemble d’oscillateurs. Un oscillateur est basé sur un modèle simple, composé de deux neurones, N_1 et N_2 [Revel and Andry, 2009]. La fréquence des oscillateurs dépend de trois paramètres α_1 , α_2 et β :

$$N_1(n+1) = N_1(n) - \beta N_2(n) + \alpha_1 \text{ and } N_2(n+1) = N_1(n) + \beta N_2(n) + \alpha_2 \quad (2.12)$$

Le signal de contrôle de chaque articulation est ainsi obtenu en faisant la somme pondérée des différents oscillateurs :

$$\theta_j(t) = \sum_{i=1}^n w_i^j \cdot O_i(t). \quad (2.13)$$

Avec $O_i(t)$ le signal de sortie de l’oscillateur i , w_i^j le poids représentant la contribution de l’oscillateur i au signal de contrôle $\theta_j(t)$ de l’articulation j .

Primitives visuelles basées sur des neurones sensibles à la direction du mouvement

En neurobiologie, des enregistrements de cellules provenant des aires V1 et MT du cerveau ont montré que des neurones de V1, ainsi que beaucoup de cellules de MT, étaient sensibles à certaines directions privilégiées de mouvement. Ces neurones ont donc une composante sensible à la direction et sélectionnent ainsi une direction particulière (“direction-selective neurons”) [Movshon et al., 1992]. De plus, une petite part des neurones de MT sélectionne des configurations de directions, en répondant à des configurations plus globales de directions (“pattern selective neurons”).

Afin de simuler ces neurones, nous avons utilisé un estimateur de flot optique (une implémentation hiérarchique de l’algorithme de Horn et Schunk [Horn B. K., 1981] basé sur les travaux d’Amiaz *et al.* [Amiaz et al., 2007]). Puis, nous avons défini des neurones sélectionnant les directions de cette manière : l’activation de chacun des neurones A_i est proportionnelle à la distance angulaire entre le stimulus visuel (le flot optique) et la direction associée, pondérée par l’intensité du mouvement :

$$A_i = \exp\left(-\frac{(\beta - \beta_i)^2}{2\tau_1^2}\right) \cdot \left(1 - \exp\left(-\frac{V^2}{2\tau_2^2}\right)\right) \quad (2.14)$$

Avec β la direction du flot optique estimé, β_i la direction privilégiée du neurone i , V l’intensité du mouvement. Enfin, τ_1 et τ_2 sont les coefficients régulant la dynamique d’activation du neurone, respectivement pour la direction du mouvement et pour son intensité. Dans la suite des expériences, τ_1 et τ_2 ont été empiriquement réglés à 20, afin d’optimiser les dynamiques des neurones réagissant à l’intensité de mouvement observé.

D’autres études ont montré que ces neurones étaient réactifs pour une amplitude pouvant varier, autour de la direction privilégiée, d’environ 40 à 60 degrés [Hol and Treue, 2001]. C’est pourquoi nous avons défini 6 différentes classes de ces neurones, réagissant pour les directions privilégiées suivantes : 0° , 60° , 120° , 180° , 240° et 300° . (Notons que comme nous utilisons les coordonnées de l’image, l’axe des Y est orienté vers le sud (90°).) Pour des mouvements de grande intensité, ce type de neurone répondra avec une valeur de 1 si le flot optique est orienté dans la direction privilégiée, et son intensité diminuera graduellement pour des intensités de mouvement moins importantes, ainsi que pour des directions éloignées de celle qui est privilégiée. Enfin, chaque vecteur vitesse de chaque pixel de l’image, lesquels sont estimés par l’algorithme de flot optique, sera par la suite codé en 6 neurones, sensibles aux orientations citées. Pour une image de 640×480 pixels, on obtiendra donc $6 \times 640 \times 480$ neurones sélectionnant les directions.

Nous avons également ajouté des neurones encodant des configurations de directions (“pattern selective neurons”, PSN), en intégrant la réponse des neurones précédents (“direction-selective neurons”). Par simplicité, et afin d’assurer une réponse temps réel de l’interaction, nous en avons défini seulement 6 qui correspondent aux précédentes directions privilégiées (voir la figure 2.16-B). Afin que ceux-ci soient sensibles à la direction i du mouvement, chacun intègre l’activité de chacun des “direction-selective neurons” de la même direction i .

Une illustration de la réponse de ces PSN pour différentes directions de mouvements du bras Katana, et captée par une caméra 640×480 , est donnée figure 2.14.

Par la suite le modèle associera, justement, le codage de ces directions de mouvements aux activités motrices les produisant. Notons que ce type de résultats est en adéquation avec les résultats observés dans [Georgopoulos et al., 1986; Kakei et al., 1999] qui montrent que des

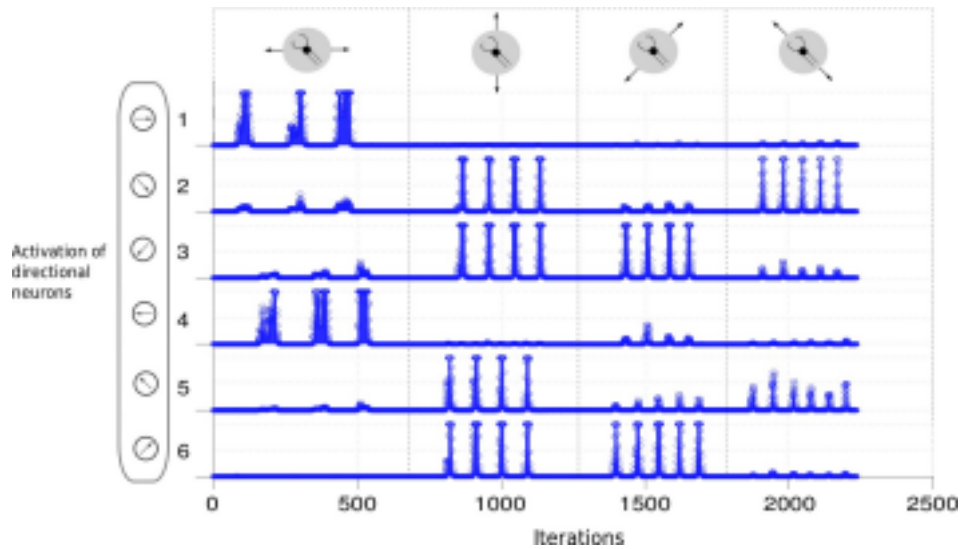


FIGURE 2.14 – Activités des neurones répondant aux directions de mouvements. En haut sont représentés les 4 types de mouvements effectués par le bras Katana face à la camera (oscillation horizontale, verticale, et deux en diagonale). À gauche sont représentées les directions privilégiées des neurones (qui correspondent aux 6 directions énumérées dans le texte).

neurones moteurs peuvent avoir une activité qui va co-varier avec des directions visuelles de mouvements.

2.3.2 Apprentissage visio-moteur et imitation bas niveau

Comme nous l'avons rappelé dans la section 2.1, il a été proposé qu'une imitation bas niveau pouvait émerger des propriétés d'un simple homéostat entre perception et action, en se basant sur une ambiguïté de la perception (voir [Gaussier et al., 1998; Rolland de Rengerve, 2013]. Poursuivant cette idée, nous avons proposé un modèle de réseau de neurones, que l'on peut voir sur la figure 2.16-A, permettant au robot, en apprenant les perceptions de ses propres mouvements, d'obtenir un comportement d'imitation d'un partenaire humain, grâce à l'ambiguïté visuelle.



FIGURE 2.15 – Système expérimental constitué d'un bras Katana dont on utilise 4 DOF, d'une caméra 640x480, et d'une personne interagissante.

Le système expérimental proposé est montré sur la figure 2.15.

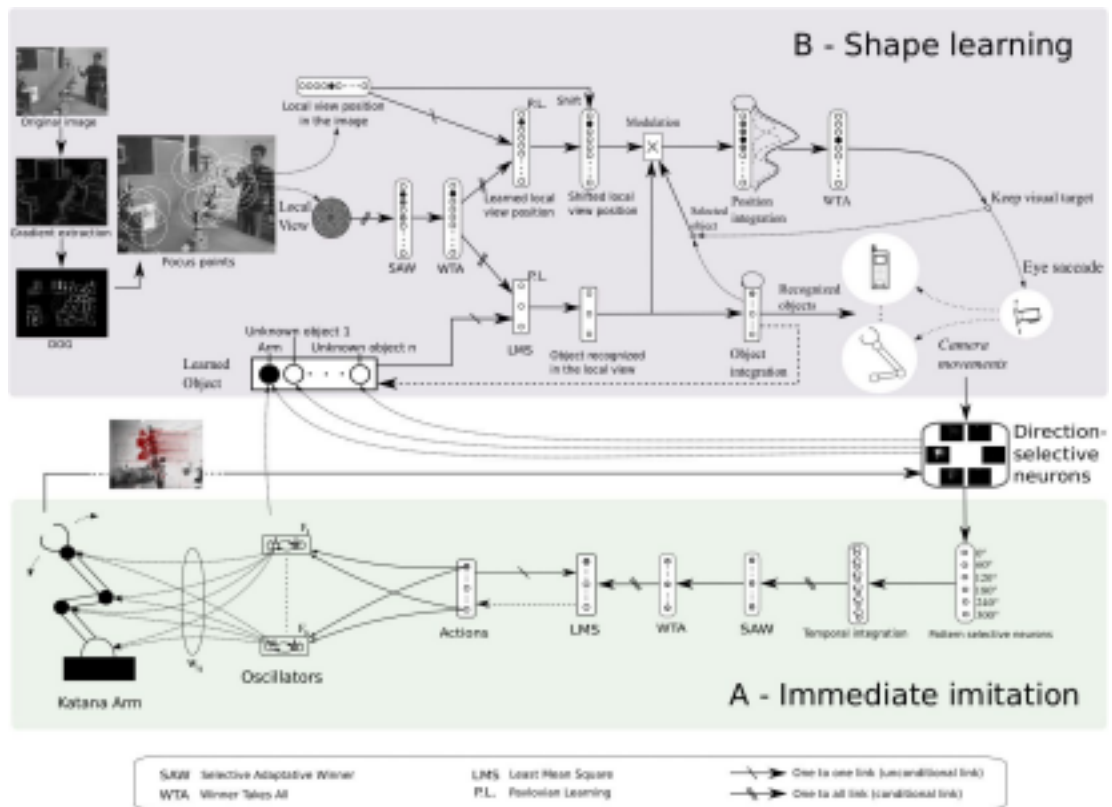


FIGURE 2.16 – Architecture globale. Partie-A : Modèle neuronal pour l'apprentissage du mouvement et l'imitation immédiate. Partie-B : Modèle neuronal pour l'apprentissage et la reconnaissance d'objets, et du bras.

Le contrôleur moteur est composé de 9 oscillateurs de 3 fréquences différentes, avec 3 décalages de phase distincts. Le signal de contrôle de chaque articulation du robot j est obtenu en additionnant les différentes sorties des oscillateurs, et en les modulant par le poids w_i^j (voir l'équation 2.13). Dans un premier essai, les poids w_i^j ont été choisis aléatoirement, le robot bougeant selon cet ensemble de paramètres. Malheureusement, même avec ce petit nombre d'oscillateurs la plupart des mouvements obtenus sont difficiles à analyser et pas vraiment biologiquement plausibles (ce qui est en partie dû aux caractéristiques mécaniques du bras Katana). Ainsi, pour plus de simplicité, nous avons dans l'expérience qui a suivi décidé de choisir les poids w_i^j permettant d'obtenir 3 actions rythmiques différentes : un déplacement horizontal (A1), et deux diagonaux (A2 et A3).

Le modèle, illustré sur la figure 2.16-A fonctionne en deux phases :

- Tout d'abord, durant une phase de babillage moteur simple, le robot apprend sa propre dynamique. Pour cela, il commence par bouger en alternant aléatoirement les trois actions A1, A2 et A3. Le "Selective Adaptive Winner" (SAW, voir section 2.1.1) a en entrée l'intégration temporelle des 6 neurones sensibles aux différentes directions de mouvements (PSN), lesquels réagissent spécifiquement à chacune des actions oscillantes observées. Selon le paramètre de vigilance du SAW, si les entrées sont suffisamment différentes de ce que les neurones encodent précédemment, alors de nouveaux neurones sont recrutés.

À la suite du SAW, un réseau de neurones de type “Winner Takes All” (WTA) est utilisé, et permet de sélectionner les neurones du SAW encodant au mieux les PSN. Pour finir, ces neurones gagnants constituent l’entrée inconditionnelle d’un réseau de neurones utilisant la règle du Least Mean Square (LMS), un perceptron simple utilisant la règle de Widrow-Hoff. Ce réseau apprend l’association entre l’encodage visuel (le flot optique) et les actions motrices représentées par l’ensemble des paramètres w_i^j .

- Après la phase d’apprentissage, lorsqu’un humain commence à bouger dans le champs visuel du robot, ses mouvements seront caractérisés par les PSN. Or, comme nous l’avons dit, le contrôleur du bras peut être vu comme étant un système homéostatique cherchant à maintenir la cohérence entre les actions produites et celles perçues, que celles-ci soient ou non exécutées par le robot lui-même ou par l’humain. Ainsi, si une configuration de mouvements induite par un humain est suffisamment proche de mouvements précédemment appris par le robot, alors le LMS lancera les paramètres d’oscillateurs w_i^j correspondants. Le robot, en lançant une action “proche” en terme de flot optique, semblera ainsi imiter le type de mouvement observé par l’humain.

2.3.3 Apprentissage de la forme du bras pendant le babillage moteur

Afin d’illustrer les possibilités d’un tel système, nous l’avons appliqué à une tâche consistant à atteindre des objets. Pour ce faire, nous allons détailler ici comment, en utilisant le mécanisme des saccades oculaires, nous obtenons une reconnaissance de la main en utilisant l’attention qui est portée sur elle. Pour cela, nous utilisons un modèle neuronal permettant la reconnaissance d’objets inspiré des travaux de [Gaussier et al., 1997] et [Lepretre et al., 2000]. Le principe général de ce modèle est d’apprendre des vues locales d’objets en se basant sur la détection de points d’intérêts (l’apprentissage successif de chacun des points attirant l’attention simule ainsi les saccades oculaires). Comme illustré dans la figure 2.16-B, le gradient spatial d’information est d’abord extrait de l’image de gris. Ce gradient est ensuite convolué à un filtre DOG (différence de gaussiennes). La sortie de ce processus fournit une carte de salience permettant de mettre en évidence les régions de l’image ayant une structure de forme proche de celle d’un coin. Enfin, les maxima locaux sont sélectionnés sur cette carte de salience.

Autour de ces points d’intérêts, des vues locales sont extraites en collectant les pixels entourant ces points (ici sur un rayon de 20 pixels). Celles-ci sont alors encodées en les filtrant par une transformation en log polaire, afin d’être robuste aux changements d’échelle et aux variations par rotations. Ces vues locales filtrées forment alors les entrées d’un autre SAW, lequel recrutera un nouveau neurone chaque fois que les entrées seront suffisamment différentes de celles rencontrées précédemment. Enfin, un réseau de type WTA est utilisé pour sélectionner les vues locales gagnantes.

Le modèle présenté dans la figure 2.16-B peut être divisé en deux parties. La reconnaissance du “what”, identifiant ce qu’est l’objet, et celle du “where”, sa localisation. Un premier algorithme de LMS est utilisé dans la voie “what” afin d’associer à chaque vue locale un objet donné. Le nombre de neurones du LMS correspond ainsi au nombre d’objet appris. Dans la voie “where”, deux LMS, pour les axes x et y , sont utilisés pour associer le centre de l’objet relativement à la vue locale. Comme représenté dans la figure 2.17, après une telle phase d’apprentissage de la voie “where”, chaque vue locale (centrée sur un point d’intérêt) propose sa

propre prédiction de la localisation du centre de l'objet auquel la voie "what" l'associe. Ainsi, le centre d'un objet sera bien positionné si la majeure partie des vues locales associées vote pour la même position. Dans le cas contraire, plusieurs positions du centre de l'objet seront prédites sans qu'une majorité de votes permette d'identifier un gagnant (voir fig.2.17). Puisque ce qui est appris est une position relative, si un objet est appris à une position donnée et détecté à une autre, la sortie du LMS décalera, pour chaque vue locale, la position relative apprise à la position courante, afin de déterminer la position courante de l'objet.

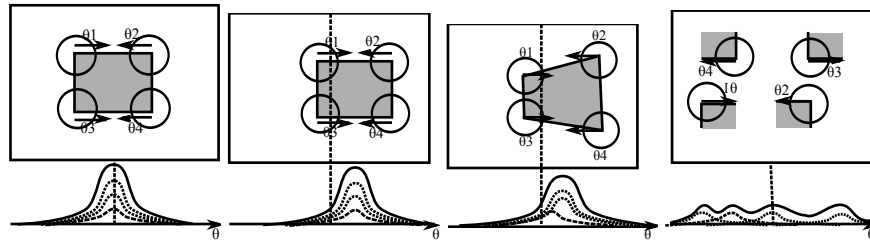


FIGURE 2.17 – Exemple schématique montrant l'estimation de la position d'un objet

Le précédent modèle est alors utilisé pour que le robot apprenne son propre bras, sans ajouter de connaissance *a priori*. Ainsi, durant la phase de babillage, le robot détecte les points d'intérêts comme décrit précédemment, et la carte de salience obtenue après le filtrage par la DOG est modulée par l'intensité du mouvement estimé (le flot optique). Ainsi, si l'on suppose que le robot percevra son bras en mouvement durant la phase de babillage, les caractéristiques détectées sur les points d'intérêts (vues locales) appartiendront, statistiquement, au bras du robot davantage que tout autre objet alentour. Puisque ses propres activités motrices, celles de son bras, génèrent ce flot optique, le robot apprendra "statistiquement" la forme de son bras.

2.3.4 Ambiguïté visuelle et émergence d'un comportement de "reaching"

Après les phases d'apprentissage (de babillage), le robot est capable de reconnaître son bras et de localiser son attention visuelle dessus. Nous avons ensuite, en plus du bras du robot, ajouté la présence d'un autre stimulus visuel susceptible d'attirer l'attention visuelle du robot. Cet objet sera également appris en utilisant le même réseau neuronal permettant l'apprentissage de formes. Il sera agité en face du robot pendant que son bras ne bouge pas, les vues locales correspondront donc statistiquement au nouvel objet.

Comme expliqué précédemment, le modèle utilisé pour la reconnaissance d'objet est aussi capable d'estimer leurs positions. Ainsi, lorsqu'un objet (ou le bras du robot) est détecté et localisé, nous pouvons utiliser cette estimation de position pour bouger la caméra pan-tilt et la centrer sur l'objet. Mais surtout, si deux objets sont reconnus simultanément dans l'image et partagent l'attention visuelle du robot, sa caméra pan-tilt bougera alternativement d'un objet à l'autre, simulant ainsi des saccades oculaires.

En conséquence, lorsque le robot perçoit son bras et centre son attention dessus, et qu'en même temps un humain lui montre un objet appris dans son champ visuel, alors la caméra commence à bouger alternativement du bras à l'objet. Or de tels déplacements, provenant du partage de l'attention visuelle, génèrent un flot optique particulier. En utilisant le modèle neuronal permettant l'imitation de l'action (voir la section 2.3.2 et la figure 2.16), le flot optique obtenu sera

codé par les neurones de directions et les PSN, lesquels, par le LMS, engageront les primitives motrices associées permettant de générer un mouvement oscillant, qui sera de fait dirigé dans la direction où l'objet a été localisé.

Les résultats expérimentaux sont illustrés à la figure 2.18. Après les phases d'apprentissage, le robot est capable de produire, et ainsi d'imiter 3 mouvements différents. Il est également capable de percevoir et de centrer son attention sur son bras ainsi que sur un objet externe (ici, le robot Aibo). La figure 2.18-A montre 3 captures d'images provenant de la caméra pendant l'expérience. Sur ces images apparaissent les vues locales (correspondant aux points d'intérêts), colorées en blanc si elles ne sont associées à aucun objet, en vert si elles sont associées au bras Katana, et en rouge si elles sont associées au robot Aibo. La vue locale "vote" pour un objet donné lorsque l'activité du LMS correspondant dépasse un seuil fixé, et le diamètre du cercle illustre la marge avec laquelle ce seuil est dépassé (les grands diamètres correspondent donc à une reconnaissance plus certaine). La figure 2.18-B montre les activités des neurones du LMS dédié à la reconnaissance des objets (verte pour le bras, et rouge pour l'Aibo). Un seuil décide de si l'objet est détecté ou non, et est représenté par la ligne noire en pointillée. La figure 2.18-C et D montre les axes X et Y de l'activité des champs de neurones (lignes bleues) et les mouvements Pan et Tilt de la caméra (ligne verte en pointillé). La figure 2.18-E montre les activités des 6 PSN. La figure 2.18-F illustre les activités des neurones du LMS dédiés à la reconnaissance des 3 actions (en lignes pointillées verte, rouge et bleu), et les lignes verte, rouge et bleu pleines montrent, quant à elles, les actions effectuées par le robot, lesquelles sont déclenchées lorsque la reconnaissance dépasse un seuil (représenté par une ligne droite en pointillé noir). Enfin, sur la figure 2.18-G et H, nous pouvons voir l'évolution de la position de l'extrémité du bras dans l'espace cartésien, calculée depuis le modèle direct du bras Katana.

Considérons maintenant le scénario proposé qui suit la période d'apprentissage, dans son ensemble. Tout d'abord, le robot ne perçoit que son bras. Celui-ci est reconnu et localisé au centre de l'image. L'objet (Aibo) est ensuite présenté par l'humain, sur le côté supérieur gauche de l'image. En conséquence, le bras et le robot sont simultanément reconnus et localisés (aire marron de la figure 2.18-B). Du fait de cette attention partagée entre son bras et un objet, la caméra se déplace vers l'objet reconnu (figure 2.18-C et D), ce qui induit un flot optique qui est encodé par les PNS (figure 2.18-E). Comme la caméra se déplace vers l'Aibo dans le coin supérieur gauche, celle-ci génère un flot optique inversé, orienté vers le bas et la droite (0° et 60° de part l'inversion de l'axe des Y). Les PNS déclenchent alors le mouvement oscillant du bras correspondant à ce qui a été appris précédemment (soit l'action 2, sur la figure 2.18-F. Le bras du robot, comme le montre la figure 2.18-G et H, se déplace alors vers l'objet détecté, vers la gauche sur l'axe des X, et vers le haut sur l'axe des Y.

2.3.5 Conclusion

Ces expériences sont basées sur l'association entre un flot optique observé et des actions permettant de produire de tels flots. Si les résultats sont convaincants pour les mouvements générant un flot relativement simple, il s'avère bien plus compliqué de catégoriser un flot optique composé de différentes directions qui s'enchaînent. De plus, il y a, du moins en théorie, une infinité de mouvements possibles combinant différentes directions atomiques, et s'il est possible de les approximer à l'aide d'une combinaison pondérée de mouvements complexes appris, ces approxi-

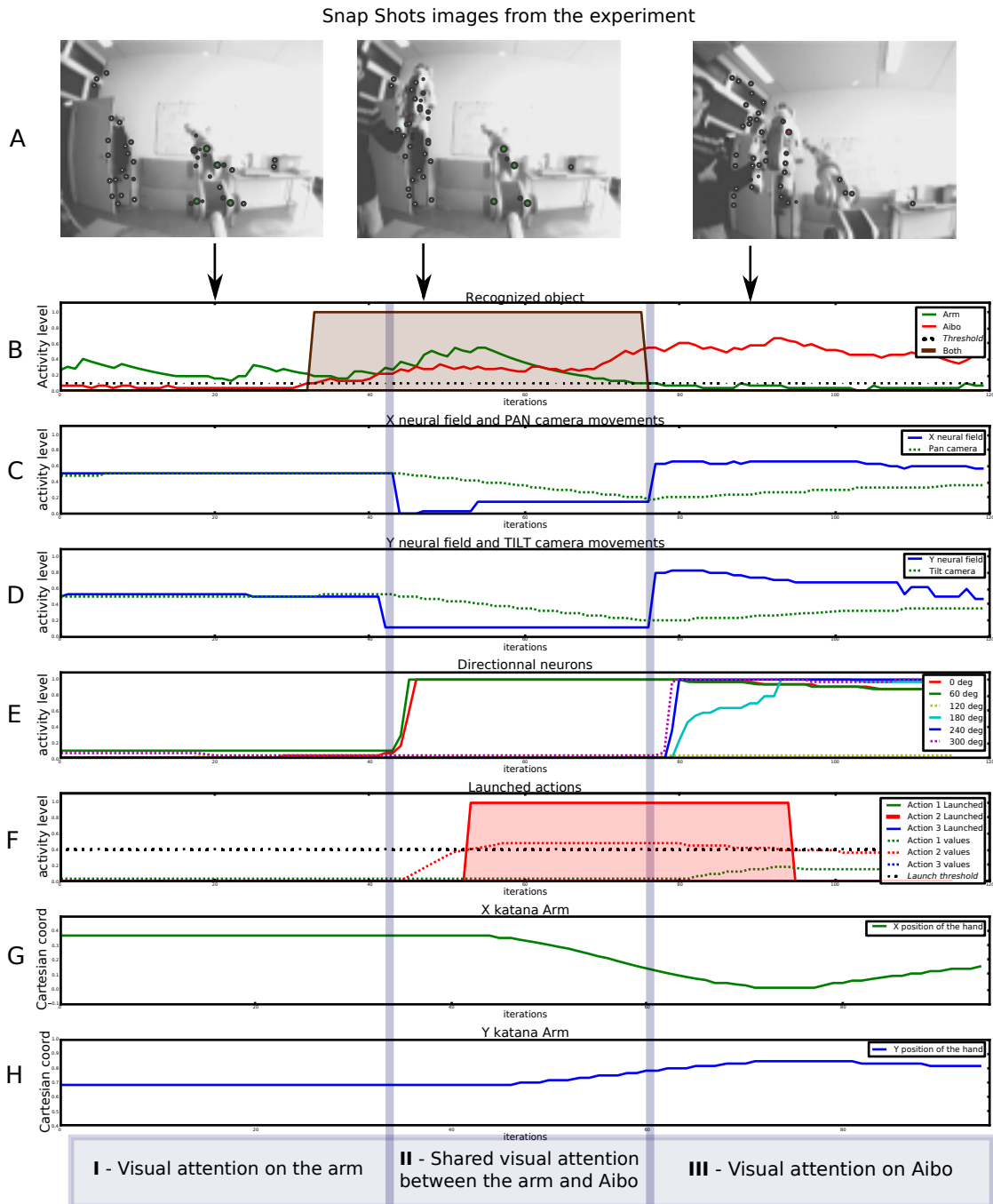


FIGURE 2.18 – Résultats expérimentaux

mations seront d'autant moins bonnes que l'apprentissage d'un seul mouvement complexe est déjà peu précis.

On pourra objecter qu'en pratique, il n'y a probablement pas une infinité de mouvements complexes qui auront à être réellement appris, si l'on considère les contraintes biomécaniques, la minimisation de l'énergie consommée, la richesse comportementale de nos primitives motrices,

ou encore le nombre limité de situations type auxquelles un enfant est confronté de fait (atteindre un objet, le saisir, le pousser, etc.). Cependant, si le nombre de mouvements complexes est de fait probablement restreint, les cas d'applications, et donc de reconnaissances possibles, sont eux effectivement infinis de par leur contexte d'application qui change chaque fois les modalités de leur application.

Ainsi, afin de pouvoir décomposer au mieux les différents mouvements observés, il semble qu'il serait plus efficace d'apprendre chaque direction atomique de manière indépendante (quand bien même une primitive motrice induit plusieurs directions atomiques). Ceci permettrait de percevoir une combinaison de flots optiques variés comme une combinaison de mouvements unitaires. Et non, comme c'est le cas dans nos expériences, de voir cette même combinaison comme un motif complexe, dont la variété de mouvements génère des caractéristiques trop peu discriminables d'un autre mouvement complexe, qui aurait un nombre non négligeable de mouvements unitaires en commun. Une telle décomposition permettrait notamment de reconnaître des combinaisons de mouvements non apprises en tant que telles, du seul fait que chaque mouvement dont cette combinaison particulière est composée serait individuellement connu et donc reconnaissable.

Le principe derrière cette idée est qu'une infinité de mouvements est possible, à partir d'un ensemble fini de mouvements. Et qu'il serait donc plus efficace d'apprendre cet ensemble fini plutôt qu'un sous-ensemble fini de combinaisons (afin éventuellement de les combiner, pour approximer de manière encore moins précise l'infinité de combinaisons possible). Notons que si d'un point de vue théorique cela peut sembler mieux, la biomécanique du corps ou l'existence de primitives motrices limite notre capacité pratique à atomiser ainsi chaque mouvement. Mais quelque soit le stade de développement d'un bébé, ou son degré de maturation, l'idée de chaque fois apprendre, séparément, les effets des commandes motrices sur lesquelles il lui est possible d'agir (quelque soit la complexité réelle de ce qu'une commande engendre), peut s'appliquer, et évoluer au cours du développement.

Notons enfin, comme nous l'avons déjà souligné, que le contexte (comme la position, l'environnement dans lequel l'enfant évolue, les objets avec lesquels il interagit, etc.) perturbe les modalités d'application des mouvements effectués, ainsi que la loi sensorimotrice permettant d'associer un mouvement à un flot optique donné. Ceci apparaît clairement lorsque l'on considère la posture dans laquelle se trouve le robot : un mouvement de l'articulation du coude ne générera pas le même mouvement selon la position dans laquelle se trouve l'épaule.

Suivant la logique que nous venons de décrire, il faudrait donc, en plus d'apprendre les mouvements relatifs à chaque flot optique unitaire, également apprendre de manière contextuelle les lois pseudo-linéaire entre moteurs, et variation de senseurs (flot optique dans le cas courant).

2.4 Conclusion

Dans ce chapitre, nous nous sommes donc intéressés à différentes manières d’encoder l’information sensorimotrice. Nous avons tout d’abord exploré une stratégie absolue en constatant son inefficience lors de l’utilisation d’outils, lié à la nécessité d’un ré-apprentissage complet lorsque le couplage de la main étendue à l’outil et de la vision échoue.

Puis nous avons proposé une architecture permettant à un apprentissage absolu d’être adapté, transformé en fonction de différents contextes parmi lesquels celui de l’utilisation d’outils. Si les résultats sont prometteurs, cette architecture, basée sur un chaînage de réseaux de neurones GF, trouve sa limite dans le fait que la plasticité qu’elle permet est dépendante du chaînage défini, lequel doit de plus être défini a priori, en dehors de toute exploration sensorimotrice spécifique au robot. Les transformations apprises, quant à elle, sont relatives à la chaîne de neurones GF qui la précède, mais aussi à un contexte donné. Or, celui-ci n’est pas appris en tant que tel, il est donc déduit quand son apprentissage pourrait permettre de prédire et donc activer la transformation associée. Dans la partie suivante, nous verrons comment la présence d’un tel apprentissage pourra également permettre au robot de “vouloir” un contexte associé à une transformation désirée (comme celle liée à l’utilisation d’outil).

Partant du principe que le robot découvre son corps par l’exploration sensorimotrice, et que ce corps peut évoluer, que ce soit à l’image du développement du bébé humain ou de part l’utilisation d’outil, nous avons pour objectif un modèle permettant d’apprendre des “transformations”, du type que nous venons de voir, de manière indépendante de tout a priori sur la mécanique du robot. Afin d’obtenir une telle plasticité nous avons, dans la dernière partie, commencé à explorer un modèle totalement relatif dans lequel des mouvements (des activités motrices dotées d’une certaine magnitude, et non des positions absolues dans un référentiel donné, comme par exemple celui de la proprioception) sont associés à une estimation du flot perçu, et donc à une variation dans l’espace sensoriel. Nous avons exploré différentes possibilités offertes par ce modèle, toutefois, les covariations apprises étant basées sur des mouvements oscillants complexes, la covariation à apprendre est le fruit de nombreuses non linéarités et réclame soit des outils d’apprentissages plus efficaces, soit un plus grand découplage des différents mouvements effectués et perçus. Enfin, soulignons que ces apprentissages sont eux aussi dépendants de contextes dans lesquels ils s’appliquent.

Dans le chapitre qui suit, nous allons nous intéresser à un modèle relatif permettant de pallier les différentes limitations que nous avons relevées dans les modèles expérimentés jusque là, tout en exploitant les avantages de ces mêmes modèles. Ce modèle sera relatif, et apprendra de manière indépendante chaque variation de capteur en contextualisant l’apprentissage à la volée.

Modèle relatif, simulations et réalisation de séquences

Dans la continuité du chapitre précédent, nous allons poursuivre notre investigation concernant l'apprentissage de magnitudes motrices liées à des mouvements dans l'espace sensoriel, tout en prenant en compte l'ensemble des limitations relevées dans les précédents modèles, à savoir :

- Être en mesure d'adapter ou de transformer de précédents apprentissages, comme par exemple dans le cas des "gain-fields", sans, pour autant, déterminer à l'avance quelles modalités influenceront quelles autres.
- Prendre en compte un contexte lié à une transformation particulière, afin de pouvoir en dernier lieu associer l'un à l'autre.
- Partir du constat qu'un grand nombre de senseurs et de moteurs sont liés dans des relations non-linéaires, et que ces relations sont elles-mêmes contextuelles : et ceci afin de tendre vers (i) un apprentissage séparé de chaque relation senseur-moteur, et (ii) une "contextualisation" de chaque non linéarité.

Dans une première partie (Sec. 3.1), nous proposerons un modèle sensorimoteur relatif prenant en compte les points (i) et (ii), par l'encodage des lois sensorimotrices pour chaque couple senseur-moteur, où chaque non-linéarité est contextualisée. Les résultats de cette manière d'apprendre seront comparés, à l'aide d'une tâche d'atteinte au cours de laquelle un outil est ajouté en milieu d'expérience, avec une stratégie absolue et une stratégie de renforcement simple.

Dans une seconde partie (Sec. 3.2) nous proposerons une manière d'exploiter un tel apprentissage de lois sensorimotrices prédictives : d'une part en présentant un mécanisme de simulation, qui simule tant l'action que le contexte, et d'autre part en illustrant l'intérêt de ce mécanisme à travers une expérience d'utilisation d'outil qui implique la génération, à la volée, d'une séquence de mouvement jamais apprise.

3.1 Modèle relatif, et comparaison avec un modèle absolu

Dans une première partie, nous allons présenter le modèle relatif que nous proposons, basé sur un encodeur de lois sensorimotrices (“Sensorimotor Law Encoder”, SLE). Nous présenterons également les modèles auxquels nous nous comparerons, à savoir un modèle issu d’une stratégie absolue, et un autre modèle servant de référence et basé sur un algorithme simple d’apprentissage par renforcement.

Dans une seconde partie, nous observerons le comportement de ces modèles lors d’un apprentissage sensorimoteur, et nous comparerons les résultats issus d’une tâche d’atteinte effectuée en simulation, au cours de laquelle un outil est ajouté et vient changer la géométrie du robot.

3.1.1 Modèle relatif prédisant des variations sensorielles

Nous allons commencer à explorer, dans cette première partie, l’idée selon laquelle toutes variations, et en premier lieu celles du monde environnant, perçues par les senseurs, sont dues à une activation motrice du robot (étant entendu qu’en réalité le monde extérieur est également sujet à des variations qui ne sont pas le fait du robot).

La première partie concerne donc le SLE, qui apprend à coder une variation de senseur *comme* une activité motrice, en fonction d’une catégorisation de senseurs. Le SLE apprend à prédire des variations sensorielles en se basant à la fois sur une activité motrice, et sur un contexte sensoriel.

Afin d’effectuer des tâches de manière “open-ended”, et que l’on pourrait qualifier de haut niveau, nous avons vu en section 1.2.6 que de nombreux auteurs défendent l’idée qu’un robot doit avoir la capacité de pouvoir apprendre des compétences ré-utilisables dans de nouveaux scénarios non expérimentés, et donc, en quelque sorte, de combiner des compétences de base pour réaliser de nouvelles tâches. Nous proposons donc que le robot n’apprenne que les tâches les plus simples (ce qui dans notre modèle, se traduit par une capacité à contrôler un unique senseur), et développe parallèlement la capacité de combiner ces différentes compétences à la volée, en fonction des contraintes liées à la tâche ou au contexte.

Suivant le principe idéomoteur (voir sec. 1.1.3), notre objectif est ainsi d’apprendre à “percevoir” des variations sensorielles comme étant équivalentes à des commandes motrices. Nous proposons une stratégie relative (voir sec. 1.2.1) dans laquelle le SLE apprend à prédire les variations sensorielles dS en fonction de la magnitude M des commandes motrices, en supposant pour cela qu’il est possible d’approximer des relations localement linéaires entre elles (nous verrons les équations dans les sections qui suivent).

Considérant que notre modèle tente de réduire dS à M , il est ici supposé par un tel modèle que les changements observés dans l’environnement du robot sont tous potentiellement le résultat de ses propres actions (voir [Pierce and Kuipers, 1997; Olsson et al., 2006] pour des travaux proposant des solutions alternatives à ce problème).

En réduisant l’apprentissage sensorimoteur au seul apprentissage des variations observées lors d’activations motrices, nous proposons d’explorer, dans ce chapitre et le suivant (chapitre 4), l’idée selon laquelle toute variation observable est due à une activation motrice du robot. Et en premier lieu, dans ce chapitre, comment toute variation du monde environnant, telle que perçue par les senseurs, peut être due à une activation motrice du robot.

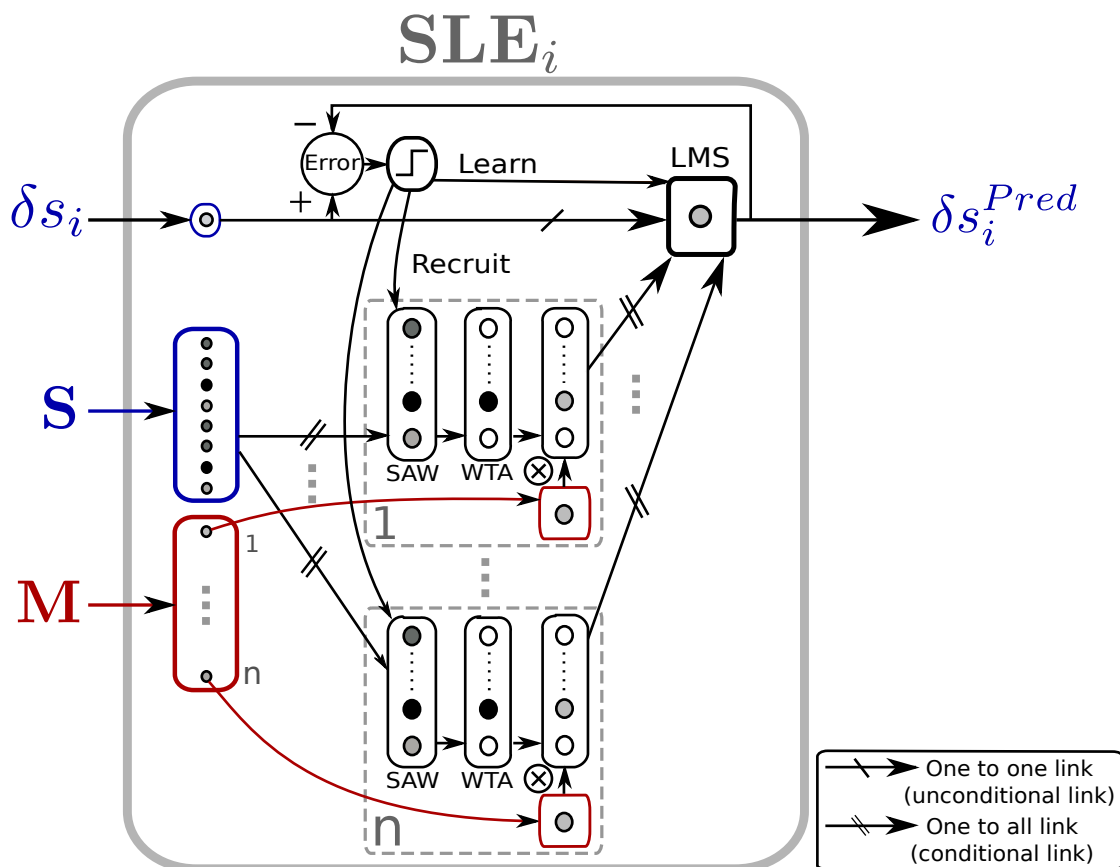


FIGURE 3.1 – SLE (“Sensorimotor Law Encoder”) du capteur i . Un prédicteur qui apprend les variations du capteur i , en prenant en entrée les vecteurs S et M de tous les capteurs et moteurs, respectivement. Une catégorisation de S , propre à chacun des n moteurs, est faite pour encoder des “contextes”. Le recrutement d’un nouveau contexte est basé sur l’erreur de prédiction de δs_i^{Pred} . Le contexte gagnant (le plus adapté) est ensuite multiplié par la magnitude du moteur correspondant, dans un neurone. Ce neurone est alors associé avec la prédiction de la variation du capteur i . La variation prédite totale (ici δs_i^{Pred}) est la somme des contributions de chaque moteur.

Cette hypothèse nous permettra de défendre l’idée qu’il n’y a pas de séparation *a priori* entre le corps du robot, et le reste du monde. Et c’est cette considération qui sera utilisée pour parvenir à faire percevoir, sans ajouter de mécanisme spécifique, l’outil comme étant une extension du corps. C’est pourquoi notre vecteur des capteurs en entrée, S , est constitué tant de capteurs relatifs aux données internes (par exemple la proprioception) qu’externes (issues par exemple de la vision du monde extérieur) au robot lui-même.

Le prédicteur SLE s’appuie sur les travaux inauguraux de Bullock et collègues [Bullock et al., 1993; Srinivasa and Grossberg, 2008; Srinivasa et al., 2012], mais comporte toutefois 3 différences fondamentales permettant de contourner les difficultés de leur modèle.

Tout d’abord, si dans leur modèle un seul prédicteur apprend l’ensemble des variations sensorielles, nous proposons dans notre modèle, en réponse au point (ii) évoqué en introduction de ce chapitre, d’apprendre autant de SLEs que de capteurs, afin de pouvoir agir indépendamment

sur chacun des senseurs auxquels les SLEs correspondent, ce qui permet notamment de générer de nouvelles combinaisons de variations. Cela permet notamment de répondre aux limitations évoquées en section 2.3.5. (Notons par ailleurs que, si l'on considère le modèle dans son ensemble, tel qu'il sera développé dans ce chapitre, les raisons sont proches de celles évoquées en sec. 1.1.5.).

Ensuite, nous proposons que chaque SLE puisse potentiellement avoir tous les senseurs en entrée. Si tous les senseurs ne sont pas nécessairement pertinents pour prédire la variation d'un seul senseur, dans le cadre d'un développement autonome des entrées inattendues peuvent se révéler être des prédicteurs clés. Par exemple, dans l'une de nos expériences, des informations tactiles seront utilisées pour prédire une variation de la position perçue de la main robotique (avec un outil). L'information sensorielle tactile n'est pas une information qu'il faut avec évidence prendre en compte pour prédire la position de la main (de manière classique, on se contente généralement de la valeur de chaque articulations, voir sec. 1.2.1), ce qui illustre qu'au cours du développement, des entrées *a priori* non pertinentes peuvent être utilisées pour apprendre des lois sensorimotrices, en raison de circonstances qui peuvent être imprévisibles. Ce faisant, nous suggérons également qu'au cours du développement, une loi erronée, imprécise ou incomplète pourrait être apprise (en l'occurrence, dans l'exemple précédent l'information tactile en soi n'est probablement pas suffisante, et requiert également la reconnaissance d'outil par la vision), ce qui est en accord avec la perspective d'un apprentissage développemental [Smith and Thelen, 2003].

Enfin, la dernière différence de notre modèle réside dans le fait que le SLE n'apprend pas de modèle inverse : ce point sera développé ultérieurement (en section 3.2).

Chaque SLE est donc dédié à un unique senseur dont il prédit la variation δi , pour le senseur i , en fonction des magnitudes motrices et de différents encodages du contexte sensoriel courant. \mathbf{M} est une copie du vecteur des N magnitudes motrices en cours d'exécution : chaque moteur m_n peut potentiellement prédire une variation de s_i en fonction du contexte donné par \mathbf{S} , le vecteur des senseurs d'entrée. Comme décrit dans la fig. 3.1, les N moteurs sont traités de la manière suivante. Dans une première étape, pour chaque moteur, \mathbf{S} est chaque fois catégorisé à l'aide d'un réseau neuronal appelé "Selective Adaptive Winner" (SAW). La règle d'activation des neurones du SAW est donnée par :

$$A_k = 1 - \frac{1}{N} \sum_j |w_{kj} - e_j| \quad (3.1)$$

où A_k est l'activité du k^{me} neurone de la couche de sortie du SAW, w_{kj} le poids de la connexion entre un neurone de sortie k et le j^{me} neurone de la couche d'entrée dont l'activité est e_j . Le poids de la connexion est fixé à e_j lorsque le recrutement intervient (ligne 7 de l'Algorithme 1). Bien qu'il n'y ait pas d'adaptation dans nos expériences, une règle permettant une adaptation pourrait être ajoutée selon le même mécanisme que celui décrit dans la section 2.1.1.

Considérant l'équation 3.1, plus les valeurs d'entrées sont proches de celles apprises, plus l'activité A_k sera élevée. La manière dont le recrutement s'effectue est décrit plus précisément aux lignes 6 à 8 de l'Algorithme 1. Notons que pour qu'un recrutement ait lieu, deux conditions doivent être remplies : une erreur est détectée par le SLE du senseur i , et un unique moteur, appelé ici na , a été activé depuis la précédente vérification effectuée par le SLE (soit le moment où une valeur de référence appelée s^{Stored} est stockée). La nouvelle catégorie, qui apprend le contexte courant, est alors recrutée dans le SAW du senseur i moteur na .

La seconde étape consiste, pour chaque moteur, en la sélection de la catégorie sensorielle la plus appropriée du SAW à l'aide du réseau de neurones "Winner Takes All" (WTA). Chaque neurone gagnant est alors multiplié par valeur de l'activité motrice correspondante, m_n . Enfin, chacun de ces N neurones, pour chaque moteur, est utilisé comme signal conditionnel pour apprendre la relation linéaire entre le moteur m_n et le signal d'entrée inconditionnel δs_i . Ceci est effectué grâce à l'apprentissage du poids dans l'algorithme des moindres carrés moyens ("Least Mean Square", LMS), dont on peut voir l'équation ligne 13 de l'Algorithme 1. Notons que comme pour le SAW, si cet algorithme est dédié à une adaptation progressive de ses différents poids, il n'est ici utilisé que pour un apprentissage en un coup. On peut également voir l'équation de la loi d'activation ligne 16 de l'Algorithme 1. Enfin, les contributions de chaque activité motrice sont additionnées dans le LMS, ce qui permet d'obtenir la variation prédite totale δs_i^{Pred} (voir l'équation ligne 17 de l'Algorithme 1).

Notons que le signal d'apprentissage venant de l'erreur de prédiction, est commun pour le recrutement du SAW et pour l'apprentissage du LMS. De plus, le déclenchement de ce signal ne dépend pas de l'espace des senseurs d'entrée, mais seulement d'un seuil Th_1 sur l'erreur de prédiction de chaque SLE (voir ligne 5 de l'Algorithme 1, et la fig. 3.1). Par ailleurs, pour éviter le bruit des mesures instantanées des dérivées des senseurs, l'erreur n'est pas mesurée à chaque itération mais seulement lorsque le senseur qui est prédit a suffisamment changé (sur la base d'un seuil arbitraire Th_1^{Step}). Si ce seuil est trop bas, le SLE pourrait être trop sensible au bruit de certains senseurs d'entrée et recruter trop de neurones. Si le seuil est trop élevé, l'apprentissage pourrait ne pas être suffisamment précis. Notons aussi, en ce qui concerne l'apprentissage du LMS, que la valeur Δs_i^{Real} utilisée est une variation qui peut potentiellement avoir été intégrée sur une période de temps plus ou moins longue, dans la mesure où l'apprentissage a lieu uniquement lorsque cette valeur atteint un certain seuil.

L'ensemble des mécanismes du SLE, pour le senseur i , est montré dans l'Algorithme 1. \mathbf{S} est le vecteur des valeurs s_i de tous les senseurs. \mathbf{M} est le vecteur des magnitudes motrices m_n de tous les moteurs. M_n est une accumulation des magnitudes motrices du moteur n . Le SLE du senseur i effectue une prédiction s_i^{Pred} de la variation d'un senseur i . Le prédicteur SLE initialise la valeur s_i^{Stored} à celle de s_i^{Real} , l'utilisant ainsi comme une mémoire à court terme de la valeur réelle. Étant donné un seuil arbitraire Th_1^{Step} , choisi empiriquement, lorsque le senseur varie suffisamment $|\Delta s_i^{Real}| > Th_1^{Step}$, ou lorsque la valeur prédite se déplace suffisamment $|\Delta s_i^{Pred}| > Th_1^{Step}$, le SLE vérifie si la prédiction est, ou n'est pas, suffisamment bonne (sur la base d'un autre seuil arbitraire, plus petit, que nous avons défini empiriquement comme $Th_1^{Err} = \frac{Th_1^{Step}}{2}$). Si c'est le cas, le prédicteur est alors réinitialisé (ligne 11). Si, de plus, la prédiction obtenue n'est pas suffisamment bonne (si $|\Delta s_i^{Pred} - \Delta s_i^{Real}| > Th_1^{Err}$), alors il y a un apprentissage, à condition que le robot n'ait eu qu'un seul moteur activé à la fois, nommé na , depuis la dernière réinitialisation (sinon on ajouterait ici le problème de l'attribution de cette variation). En ce cas, la sous-partie concernant le moteur na du SLE recrute un nouveau neurone dans le SAW^{na} , et apprend le poids associé dans le LMS^{na} (ligne 6 et 13). Cette méthode permet de prédire la variation réelle observée Δs_i^{Real} . Notons que cet apprentissage est ainsi contextualisé en fonction de la configuration des senseurs \mathbf{S} , que ces contextualisations sont propres à chaque moteur étant à même de faire varier le senseur concerné, et que la prédiction est linéaire avec la magnitude de chaque moteur. A chaque itération, pour chaque moteur, la magnitude du moteur (ligne 14) et la variation prédite du senseur (ligne 15, 16 et 17) sont accumulées jusqu'à l'étape d'initialisation suivante. Notons l'utilisation de variables temporaires,

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

```

1:  $s_i^{Stored} \leftarrow s_i^{Real} ; \Delta s_i^{Pred} \leftarrow 0 ; \forall n, M_n \leftarrow 0$ 
2: loop
3:    $\Delta s_i^{Real} \leftarrow s_i^{Real} - s_i^{Stored}$ 
4:   if  $|\Delta s_i^{Real}| > Th_1^{Step}$  or  $|\Delta s_i^{Pred}| > Th_1^{Step}$  then
5:     if  $|\Delta s_i^{Pred} - \Delta s_i^{Real}| > Th_1^{Err}$  then
6:       // Error : Recruitment for the (unique) activated motor  $na$ , i.e. where  $M_n \neq 0$ , of the  $k^{th}$ 
       // neuron in  $SAW^{na}$ , and learning in  $LMS^{na}$ 
7:        $\forall j, w_{kj}^{SAW^{na}} \leftarrow s_j^{Real}$ 
8:        $w_k^{LMS^{na}} \leftarrow \frac{\Delta s_i^{Real}}{M_{na}}$ 
9:     end if
10:    // New step : re-initialisation of the predictor
11:     $s_i^{Stored} \leftarrow s_i^{Real} ; \Delta s_i^{Pred} \leftarrow 0 ; \forall n, M_n \leftarrow 0$ 
12:  end if
13:  // Perform current prediction
14:   $\forall n, M_n \leftarrow M_n + m_n$ 
15:   $\forall n, \begin{cases} \forall k, A_k^n \leftarrow 1 - \frac{1}{N} \sum_j |w_{kj}^{SAW^n} - e_j| \\ \forall k, \text{ if } k = \arg \max_k (A_k^n), A_k^n \leftarrow 1, \text{ else } A_k^n \leftarrow 0 \\ \forall k, B_k^n \leftarrow m_n A_k^n \end{cases}$ 
16:   $\delta s_i^{Pred} \leftarrow \sum_n \sum_k w_{kn}^{LMS^n} B_k^n$ 
17:   $\Delta s_i^{Pred} \leftarrow \Delta s_i^{Pred} + \delta s_i^{Pred}$ 
18: end loop

```

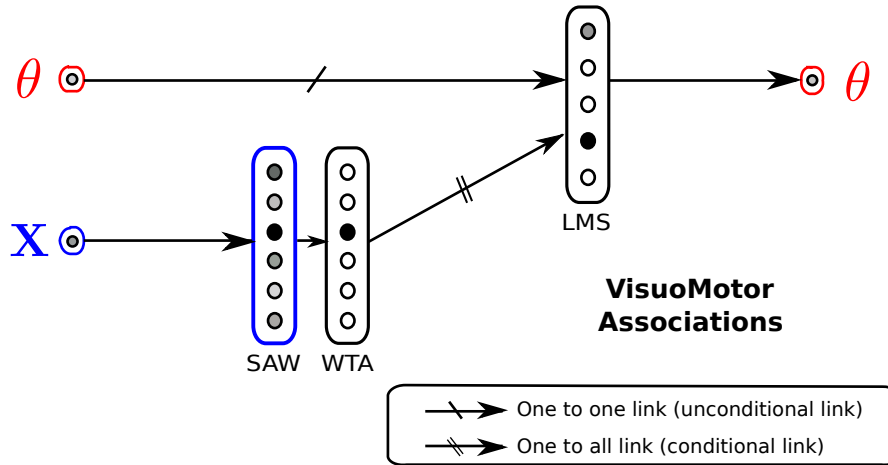


FIGURE 3.2 – Modèle “absolu” couplant des positions dans l’espace visuel à des positions articulaires

les vecteurs A et B , ligne 15, permettant de calculer les valeurs du SAW dans le vecteur A , de les mettre à jour à la suite du WTA, et d’obtenir le vecteur B qui suit la multiplication par la matrice motrice. Ce vecteur est alors utilisé en entrée du LSM et permet de calculer δs_i^{Pred} (voir ligne 16).

Ainsi, à tout moment, grâce à ce mécanisme de prédiction permettant d’obtenir δs_i^{Pred} en fonction de M , il est possible de déduire une prédiction de la position absolue du capteur correspondant, en se basant sur la valeur courante du capteur ($s_i(t)$) et les matrices motrices courantes, en effectuant :

$$s_i^{pred}(t+1) = s_i(t) + \delta s_i^{pred}(t) \quad (3.2)$$

3.1.2 Modèle “absolu” utilisé pour la comparaison

Dans cette section, nous allons détailler un modèle qui suit une stratégie "absolu". Globalement, il s’agit d’une méthode d’apprentissage par des RBF (fonctions de bases radiales). Cela nous permettra de comparer l’encodage effectué par ces différentes stratégies, ainsi que les différentes performances lors d’une tâche d’atteinte.

Dans le modèle absolu, nous traiterons le vecteur des positions visuelles X (ici les capteurs) et celui des positions angulaires des articulations du robot, θ (ici les moteurs). La configuration sensorielle sera catégorisée, et associée à une configuration motrice donnée. Après l’apprentissage, une telle association permet au modèle d’activer la configuration motrice correspondant au mieux aux entrées sensorielles.

L’algorithme est donc composé d’une couche de catégorisation de l’espace des capteurs, d’une couche qui sélectionne la catégorie gagnante, et enfin d’un algorithme d’apprentissage qui associe une sortie de matrice associée aux valeurs des capteurs (voir fig. 3.2).

Le réseau SAW, précédemment décrit, est ici utilisé pour la catégorisation (voir eq. 3.1). Cependant, ici la règle de recrutement est fonction d’une valeur *vigilance*, qui permet de recruter chaque fois qu’aucun neurone du réseau n’est suffisamment activé, c’est-à-dire lorsque les

entrées sensorielles sont trop éloignées de celles déjà rencontrées et apprises par ce réseau. Les équations sont les suivantes :

$$A_k = 1 - \frac{1}{N} \sum_j |w_{kj} - e_j| \quad (3.3)$$

$$\Delta w_{kj} = \epsilon^{SAW} (e_j - w_{kj}) \quad (3.4)$$

$$\epsilon^{SAW} = \begin{cases} 1 & \text{if } A_k > \text{vigilance} \\ 0 & \text{else} \end{cases} \quad (3.5)$$

Avec A_k l'activité du k^{ime} neurone de la couche de sortie, ϵ le taux d'apprentissage, et w_{kj} le poids de la connexion entre un neurone de sortie k et le j^{ime} neurone de la couche d'entrée, dont l'activité est e_j . La *vigilance* est le seuil de recrutement qui détermine la granularité des états dans l'espace proprioceptif.

Cette étape de catégorisation est suivie d'un réseau neuronal "Winner Takes All" (WTA). Le neurone gagnant apprend alors la sortie souhaitée, c'est-à-dire la configuration articulaire θ , grâce à un algorithme d'apprentissage basé sur l'algorithme de Least Mean Square (LMS). L'activité de sortie est donnée par les équations suivantes :

$$O_i = \sum_k W_{ik} \cdot e_k \quad (3.6)$$

$$\Delta W_{ik} = \epsilon^{LMS} \cdot (\theta_i - O_i) \cdot e_k \quad (3.7)$$

$$\epsilon^{LMS} = \epsilon^{SAW} \quad (3.8)$$

Avec O_i la sortie du LMS, e_k l'entrée provenant du WTA. La règle d'apprentissage adapte les poids W_{ik} afin de faire correspondre la sortie avec le stimulus inconditionnel désiré θ_i . Par souci de simplicité, nous n'utilisons ici qu'un apprentissage unique sans adaptation, pour le SAW comme pour le LMS ($\epsilon^{LMS} = \epsilon^{SAW} = 1$).

Cet algorithme, suivant une stratégie absolue, peut ici être considéré comme un système homéostatique (voir sec. 2.1.1). Après l'apprentissage des catégories visuelles, même en présence de bruits ou de changements, tout conflit entre les entrées visuelles du robot et la configuration proprioceptive associée forcera le système à stabiliser sa sortie dans la configuration apprise, l'amenant à bouger pour atteindre cette position. Si ce mécanisme n'est utilisé ici qu'avec des entrées visuelles, les catégories apprises pourraient, par exemple, former des attracteurs visio-moteurs utilisant à la fois des entrées visuelles et proprioceptives, voir [Gaussier and Zrehen, 1995; De Rengervé et al., 2015].

Notons que dans la suite de nos travaux, lorsque nous nous référerons à la stratégie dite "absolue" il sera fait référence à ce modèle, reposant de fait sur l'apprentissage de RBF.

3.1.3 Comparaison de l'encodage de l'information effectué par les différentes stratégies

Encodage de l'information par la stratégie absolue

Afin de comparer les modèles absolu et relatif, nous avons commencé par observer ce qui se passe au cours de l'apprentissage dans chacun des cas. Pour cela, nous avons commencé par

simuler un bras doté d'un unique degré de liberté, avec une position articulaire θ , et une entrée visuelle x de la position de l'effecteur terminal du bras. Dans cette simulation, on a borné x de sorte que pour toute activation motrice, on ait $0 \leq x \leq 1$. Dans ces bornes, nous avons modélisé la loi suivante : $x = 0.5 \theta$ si $\theta \leq 0.5$ et $x = 1.5 \theta$ si $\theta > 0.5$.

Pour voir comment cette stratégie absolue (AS) encode l'information, nous observons comment les neurones encodent les informations visuelles et articulaires ensemble.

Dans la figure 3.3 a), nous pouvons observer, pendant une phase de babillage moteur, la position réelle (fournie en entrée du simulateur) de l'entrée visuelle, ainsi que la position encodée par le modèle AS. Dans la figure 3.3 c), nous montrons le nombre total de neurones recrutés dans le temps.

Dans AS (en bleu), les neurones sont recrutés chaque fois que l'entrée visuelle est suffisamment éloignée de la position encodée par tous les autres neurones (voir équation 3.5). Puisque notre babillage va et vient de 0 à 1, ils sont tous recrutés une fois que l'entrée visuelle atteint 1 pour la première fois, l'espace des senseurs ayant été entièrement parcouru. Les neurones du SAW catégorisent donc la position dans l'espace sensoriel lors de leur recrutement. La présence du réseau WTA, qui suit la couche de catégorisation, explique pourquoi dans cette implémentation de l'AS, les positions encodées varient pas à pas et non graduellement, car seul le centroïde de chaque catégorie est appris. Nous pouvons toutefois noter que ce pas peut être aussi petit que nous le désirons, dans la mesure où il suffit d'augmenter le seuil de vigilance dans le SAW pour recruter plus de catégories.

Encodage de l'information par la stratégie relative

De la même manière que pour la stratégie absolue (AS), nous avons effectué les mêmes expériences avec la stratégie relative (RS) basée sur le SLE.

Dans la figure 3.3 b) nous pouvons voir, durant une phase de babillage moteur la position réelle de l'entrée visuelle en entrée du simulateur, ainsi que la position codée par le modèle RS. Dans la figure 3.3 c), nous montrons le nombre total de neurones recrutés au cours du temps.

Dans le SLE, les neurones sont recrutés chaque fois que l'erreur de prédiction est au-delà d'un seuil donné. Mais, comme chaque neurone encode la loi linéaire entre l'activité motrice et la variation de capteur associée, le nombre total de neurones recrutés ne dépend pas du seuil d'erreur mais seulement de la non-linéarité de la loi qui doit être apprise. Chaque neurone code donc une loi sensorimotrice, qui est appliquée localement, en fonction du contexte gagnant donné par le neurone le plus activé du SAW.

Dans notre exemple, la loi entre x et θ est linéaire entre $[0, 0.5]$, non linéaire à $\theta = 0.5$, puis de nouveau linéaire entre $[0.5, 1]$. Elle est également non linéaire à 0 et 1 puisque $0 \leq x \leq 1$. Donc, théoriquement, puisqu'il existe trois non linéarités différentes, au minimum trois neurones devraient être recrutés. Dans cette expérience, nous pouvons observer que le premier neurone est recruté lorsque $x = 0.1$. Sur la figure 3.3 b), le trait noir pointillé représente la valeur réelle du senseur, et le trait vert plein la valeur prédite : ainsi, il y a des erreurs chaque fois que ce dernier s'éloigne de la valeur réelle. Notons que nous avons aussi reporté l'évolution des lois apprises à chaque étape de recrutement, sur la figure 3.4. La prédiction initiale était $x^{pred} = 0$ puisque le SAW n'a aucun neurones recruté au début du babillage moteur. Lorsque ce dernier commence, l'erreur entre x et x^{pred} dépasse le seuil Th_1^{Step} (fixé à 0.1 dans notre exemple). Le SAW recrute alors un nouveau neurone qui code pour une loi linéaire entre $\hat{\theta}$ et \dot{x} . Puisqu'il y a alors un unique neurone dans le SAW, il est toujours gagnant en sortie du WTA, et la loi associée se trouve ainsi

généralisée. Ce seul neurone est donc suffisant pour coder sans erreur la loi lorsque le babillage moteur est effectué entre $\theta = 0$ et $\theta = 0.5$ (où la loi linéaire est la même).

La prochaine erreur apparaît donc après $\theta = 0.5$ (et dans notre exemple figure 3.3 b), quand $\theta = 0.65$) car alors l'erreur de prédiction dépasse le seuil Th_1^{Step} , la prédiction étant elle à $\theta^{Pred} = 0.55$. Un nouveau neurone est donc recruté pour encoder la nouvelle loi. Et de la même manière, un nouveau neurone est également recruté lorsque le capteur atteint 1, parce qu'alors le bras est bloqué à son extrémité, et aucun mouvement n'est de fait effectué.

Cependant, l'apprentissage ne s'arrête pas ici, c'est à dire après avoir parcouru une première fois l'espace des senseurs, comme c'est le cas avec la AS. Or les non-linéarités de la loi sont détectées lorsque la loi linéaire change : c'est donc aux frontières d'applications d'une nouvelle loi, qui sera localement linéaire, que les neurones du SAW sont recrutés, et non au centroïde du contexte sensoriel dans lequel elle s'applique, comme ce serait pourtant optimal vu leur activation locale (notons que cette remarque importante elle sera développée dans le chapitre suivant, chap. 4).

Ainsi, les frontières émergentes des lois apprises, données par le SAW et le WTA, après le recrutement, ne sont pas en adéquation avec les frontières de la loi sensorimotrice réelle. De ce fait il faut recruter davantage de neurones pour définir plus précisément ces frontières. Dans notre exemple, quand l'effecteur exécute exactement le même mouvement la deuxième fois (second aller retour de 0 à 1 sur la figure 3.3 b)), le premier neurone (encodant $x = 0.1$) est gagnant jusqu'à ce que le deuxième (encodant $x = 0.65$) gagne, soit lorsque $x = \frac{0.1+0.65}{2} = 0,375$. Après ce point, la seconde loi linéaire avoir été recrutée est utilisée par le prédicteur SLE, alors que c'est en fait la première loi qui devrait toujours s'appliquer en réalité. C'est pourquoi un autre neurone est recruté pour corriger cette erreur, encodant la même loi que la première.

Nous pouvons observer plus précisément comment ce mécanisme fonctionne à l'aide de la figure 3.4. En supposant une activité motrice arbitraire de $\dot{\theta} = 1$, nous observons la prédiction qui serait faite par le SLE en tout point de la proprioception (θ). Puisque ces courbes donnent les variations prédites des positions de l'entrée visuelle dans tous les contextes possibles, elles donnent la loi sensorimotrice qui a été apprise. Cette loi devrait donc être $+0.5$ quand $0 \leq \theta \leq 0.5$, $+1.5$ quand $0.5 \leq \theta < 1$ et 0 quand $\theta = 1$, elle est représentée dans la figure 3.4 a). Nous pouvons observer l'évolution de la loi qui est apprise chaque fois qu'un nouveau neurone est recruté, et la généralisation que cela entraîne, dans la figure 3.4 b). Ceci nous montre donc comment la loi est apprise, et comment les frontières se déplacent jusqu'à ce qu'elle soit suffisamment proche (en fonction de Th_1^{Step}) de la loi réelle.

Observons maintenant les conséquences de cet encodage du point de vue de l'action que cela entraîne. Nous verrons par la suite comment permettre au robot d'agir à partir du SLE, l'idée sera d'utiliser la prédiction obtenue en cas d'action (ce que montre donc la figure 3.4), pour initier cette action lorsqu'une telle variation est désirée, c'est à dire afin de satisfaire un but donné.

Notons que, du fait de la très grande généralisation que peut entraîner un recrutement, avec un seul neurone le robot est capable de prédire un mouvement dans la bonne direction (voir l'étape 0 de la figure 3.4 b)), ce qui peut d'avérer suffisant pour justifier l'activation motrice du moteur correspondant. Toutefois, notons également qu'aux étapes 2 et 3 de la figure 3.4 b)), il sera prédit qu'aucun mouvement n'est possible autour de $0.8 \leq \theta \leq 1$, ce qui sera de fait pénalisant pour sélectionner l'action, du moins jusqu'à ce que la loi soit apprise plus précisément, aux recrutements qui suivent.

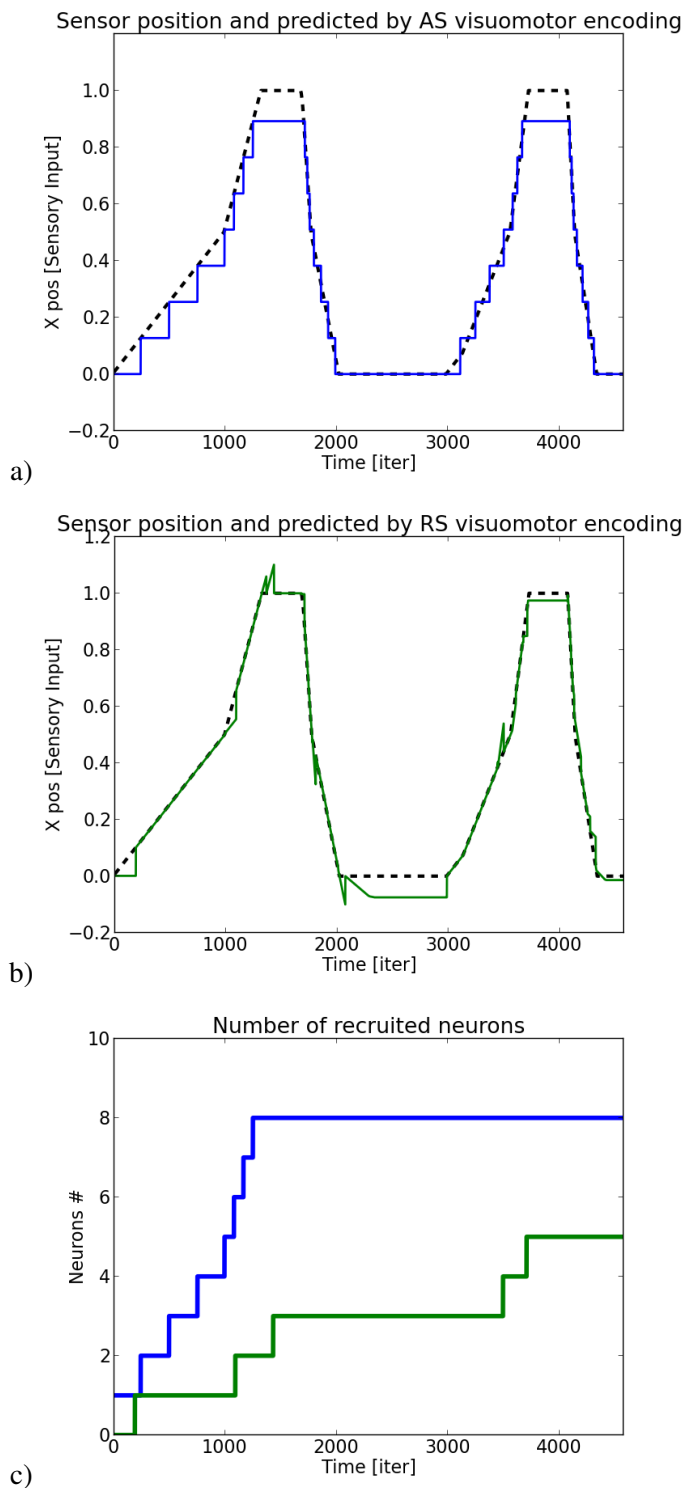


FIGURE 3.3 – Encodage de la position du capteur (la position réelle est en tiret noir), pour le modèle AS (en bleu) et le modèle RS (en vert). a) Représente les positions encodées du capteur par le modèle AS. b) Représente les positions encodées du capteur par le modèle RS. c) Représente le nombre de neurones recrutés pour les deux modèles.

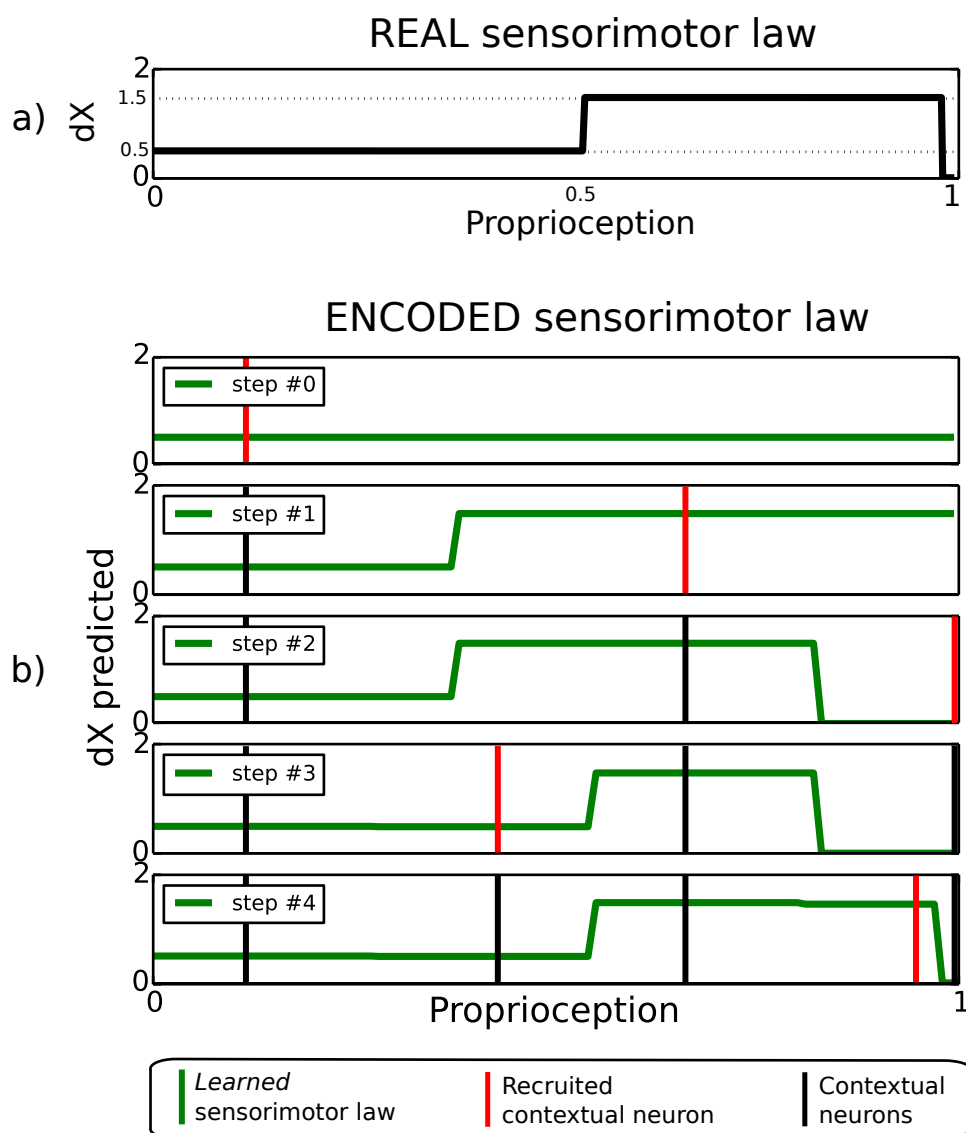


FIGURE 3.4 – En ordonnée, la quantité de mouvement du senseur \dot{x} , réel en a) et prédit en b), en fonction de la position de la proprioception θ en abscisse, dans le cas d'un mouvement arbitraire du moteur, en l'occurrence dans le cas où $\dot{\theta} = 1$. a) Loi réelle b) Loi prédite. Nous montrons ici l'évolution de cette loi en fonction du recrutement : les barres rouges indiquent un neurone du SAW nouvellement recruté, et les barres noires indiquent l'emplacement des autres neurones du SAW, chacun des neurones dépendant d'une position de la proprioception, et codant une loi donnée.

3.1.4 Expériences d'atteintes avec des stratégies “relative” et “absolue”

Si nous avons déjà expliqué comment le modèle absolu pouvait agir (en utilisant l'encodage sensorimoteur comme un homéostat qui doit se maintenir), il nous reste à expliquer ici comment le modèle relatif peut agir. Mais, tout d'abord, nous allons présenter brièvement une troisième méthode qui sera utilisée pour nos comparaisons. Celle-ci se contente d'adapter continuellement,

en fonction d'un signal de renforcement, ses chances de sélectionner ses différentes activités motrices disponibles. De par sa simplicité, ce modèle nous donnera une référence basse, que l'on attend être surpassée par les deux autres stratégies. Notons également que la simplicité de cette méthode permet de suggérer que celle-ci pourrait constituer une alternative, toujours présente, aux autres modèles en cas de défaillance de ceux-ci.

Stratégie supplémentaire : par renforcement

Dans cette stratégie de renforcement (RLS), basée sur [Sutton and Barto, 1998], un simple bruit sur les commandes motrices possibles active une configuration aléatoire des commandes $\dot{\theta}(t)$, à l'instant t . Cette configuration motrice sera alors renforcée en fonction d'un signal de récompense, noté R . Ce signal de récompense est fonction des coordonnées visuelles de l'effecteur terminal $\mathbf{X}^H(t)$, ainsi que des coordonnées visuelles de la cible $\mathbf{X}^T(t)$. Pour cette tâche d'atteinte, on récompense la diminution de la distance entre l'effecteur final et la cible. Ce modèle de renforcement suit les équations suivantes :

$$\dot{\theta}(t) = \mathcal{H}\left(\sum_{i=1}^N W_i \cdot e(t) + noise\right) = \{-1, 0, 1\} \quad (3.9)$$

$$e(t) = 1 \quad (3.10)$$

$$dW = \lambda \cdot \dot{\theta}(t-1) \cdot e(t-1) \cdot \Delta R(t) \quad (3.11)$$

$$\Delta R(t) = R(t) - R(t-1) \quad (3.12)$$

$$R(t) = |\mathbf{X}^T(t-1) - \mathbf{X}^H(t-1)| - |\mathbf{X}^T(t) - \mathbf{X}^H(t)| \quad (3.13)$$

Avec \mathcal{H} la fonction Heaviside. La configuration des activations motrices $\dot{\theta}$ est fonction des poids appris W , des entrées e fixées à 1 pour n'être conditionnées par autre chose que par l'apprentissage passé, et d'un bruit uniforme additif $noise \in [0, 1]$ (venant perturber les résultats de l'apprentissage, et dans une plus forte mesure lorsque celui-ci est peu marqué). Les poids sont adaptés avec un taux d'apprentissage λ .

Modèle relatif prédictif : comment agir

Concernant le modèle du SLE, dans la mesure où il n'y a pas d'apprentissage d'un modèle inverse, un mécanisme est requis pour agir. Dans une première approche que nous allons détailler ici (une seconde approche plus générale sera détaillée dans la partie suivante, sec. 3.2), nous avons utilisé un paradigme de type acteur-critique pour agir, utilisant un signal de renforcement prédit par des actions simulées.

Comme pour le modèle que nous venons de décrire (RLS), un signal de récompense $R(t)$ est calculé en suivant une équation similaire à celle de RS (voir l'équation 3.9). Toutefois, les équations de RS reposent sur \mathbf{X}^H et \mathbf{X}^T aux instants t et $t-1$: $\Delta R(t)$ est calculé après qu'une action a été effectuée. Mais notre méthode, elle, utilise la prédiction du SLE pour calculer une récompense prédite $\Delta R^{pred}(t+1)$. Par la suite les actions choisies pour chaque articulation θ_i suivent les équations suivantes :

$$\dot{\theta}_i(t) = \arg \max_{\dot{\theta}_i^*(t)} (\Delta R^{pred}(t+1)) \quad (3.14)$$

$$\dot{\theta}_i^*(t) = \{-1, 0, 1\} \quad (3.15)$$

$$\Delta R^{pred}(t+1) = R^{pred}(t+1) - R(t) \quad (3.16)$$

$$R^{pred}(t+1) = |\mathbf{X}_T(t) - \mathbf{X}_H(t)| - \quad (3.17)$$

$$|\mathbf{X}_T(t+1) - \mathbf{X}_H^{pred}(t+1)| \quad (3.18)$$

$$\mathbf{X}_H^{pred}(t+1) = \mathbf{X}_H(t) + \Delta \mathbf{X}_H^{pred}(t) \quad (3.19)$$

L'équation de la position $\mathbf{X}_H^{pred}(t+1)$ est une application de l'équation 3.2. Notons qu'ici nous supposons que $\mathbf{X}_T(t+1) = \mathbf{X}_T(t)$.

Ainsi, il n'y a pas d'adaptation, ni d'apprentissage de lois sensorimotrices spécifiques correspondant à une tâche donnée. Si la fonction R donnant la récompense change, le SLE pourra toujours permettre de calculer les actions à entreprendre, sans aucun nouvel apprentissage. Les actions sont mises à jour à chaque itération, jusqu'à ce que la tâche soit accomplie.

Notons que ce mécanisme fournit des commandes motrices pour, potentiellement, tous les moteurs en entrée du SLE. Or notre modèle suppose l'existence d'un SLE par senseur. Ainsi, si plusieurs SLE partagent les mêmes moteurs (dans l'expérience qui suit par exemple, il y a un SLE par axe x , y et z , et chaque SLE contient l'ensemble des commandes motrices du robot), il peut y avoir différentes commandes motrices en compétition. Il peut donc y avoir des commandes motrices avec par exemple des valeurs opposées, sur un même moteur. Un second mécanisme de sélection est donc nécessaire. Nous proposons d'utiliser un mécanisme basé sur "l'efficacité", en sélectionnant la commande motrice associée au SLE ayant un $\Delta R^{pred}(t+1)$ maximum. Ce mécanisme sera repris et plus développé dans la section 3.2.3.

Expériences d'atteintes

Pour chacun des trois modèles présenté, stratégie de renforcement (RLS), stratégie absolue (AS) et relative (RS), nous effectuons les mêmes expériences, en simultanément, dans l'espace cartésien 3D, et avec un bras à 4 degrés de liberté (DOF).

Ainsi pour obtenir une meilleure comparaison, toutes les expériences qui suivent sont effectuées en simulation afin de pouvoir les réaliser en parallèle, avec les trois modèles. Au préalable, chaque modèle a également été testé sur robot réel, avec le bras Katana que nous avons dans notre laboratoire. La simulation utilisée est celle de ce même bras robotique (voir sec. 4.3.2 pour voir les équations exactes). L'installation expérimentale utilisée pour la tâche d'atteinte peut ainsi être vu sur la figure 3.5.

Nous avons effectué une série d'expériences pour tester l'évolution des performances de la tâche d'atteinte dans le temps. Pour ce faire, les 3 robots simulés fonctionnent donc en parallèle, un par modèle. Ils font tous le même babillage moteur, durant lequel les modèles AS et RS apprennent. Régulièrement, nous arrêtons le babillage moteur ainsi que l'apprentissage, afin d'effectuer une tâche d'atteinte. Celle-ci consiste en une séquence continue de coordonnées, dans l'espace 3D, d'une cible que les 3 effecteurs des bras simulés doivent atteindre. Cette séquence est toujours la même. Elle est issue de l'enregistrement des coordonnées de l'effecteur

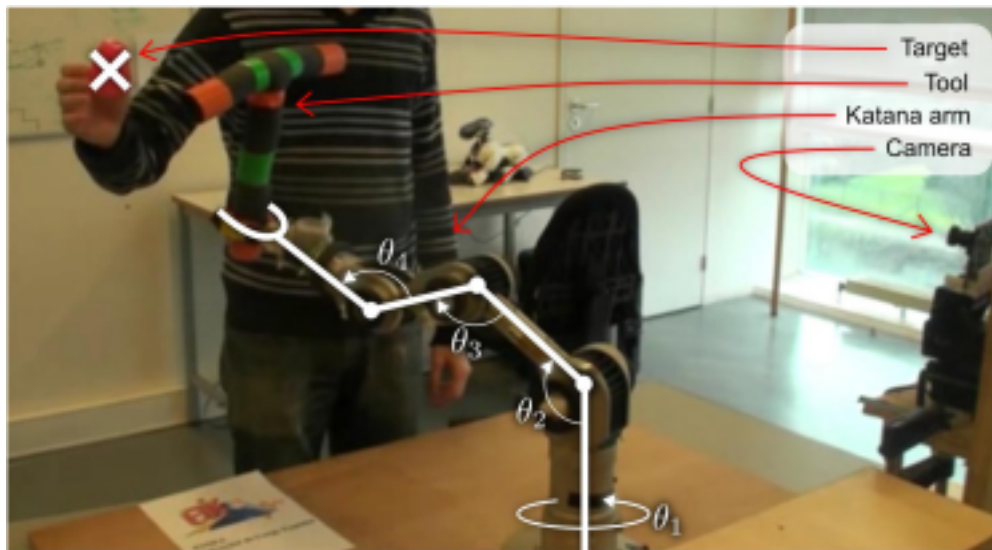


FIGURE 3.5 – Installation expérimentale réelle avec le bras katana, une caméra, un outil et une balle pour la cible. En blanc, la configuration de simulation du bras robotique avec les 4-DOF, dans l'espace 3D.

au cours d'une phase d'exploration aléatoire (effectuée précédemment). Cette tâche dure 5000 itérations. Au cours de cette tâche, la cible se stabilise pour une courte période 120 fois. À la fin de cette période de stabilisation, au moment où la cible va bouger de nouveau, nous mesurons la distance euclidienne de chaque effecteur à la cible. A la fin de la tâche d'atteinte, nous pouvons alors obtenir une valeur moyenne des distances à la cible, pour chaque modèle.

Dans la figure 3.6 a), on peut voir l'évolution de ces valeurs dans le temps. Nous notons que la stratégie de renforcement, en rouge, a des résultats assez constants, car il n'y a pas d'apprentissage. Le modèle AS diminue rapidement dans un premier temps, puis lentement lorsque la plupart de l'espace atteignable a déjà été exploré. Le modèle RS diminue lui plus rapidement que le modèle AS.

Pour tester la robustesse de ces modèles à la présence d'un outil, nous avons étendu la longueur du segment final du bras kanata simulé dans toutes les simulations. Nous étendons également les positions des coordonnées de la cible en conséquence. Nous avons effectué ce changement après que chaque modèle ait des résultats relativement stables dans la tâche d'atteinte, en l'occurrence à l'itération 21000. Les résultats sont montrés dans la figure 3.6 b). Ils indiquent clairement une robustesse de RS, alors que AS marque une nette baisse des performances juste après cette modification de la géométrie du robot. Cette différence s'explique par le fait que pour l'AS, les coordonnées de la cible sont associées à une configuration motrice du robot ayant une géométrie qui a depuis lors varié. Notons que ces résultats corroborent l'analyse préliminaire effectuée dans la section 2.1.2. Pour RS la commande dépend de la distance entre la cible et l'effecteur, laquelle prend donc en compte le changement de géométrie induit par la présence de l'outil. De plus, si la loi sensorimotrice peut légèrement varier, les directions de co-variations entre la position de l'effecteur, et les positions des articulations, restent elles globalement identiques, ce qui permet de conserver des commandes allant dans le sens désiré.

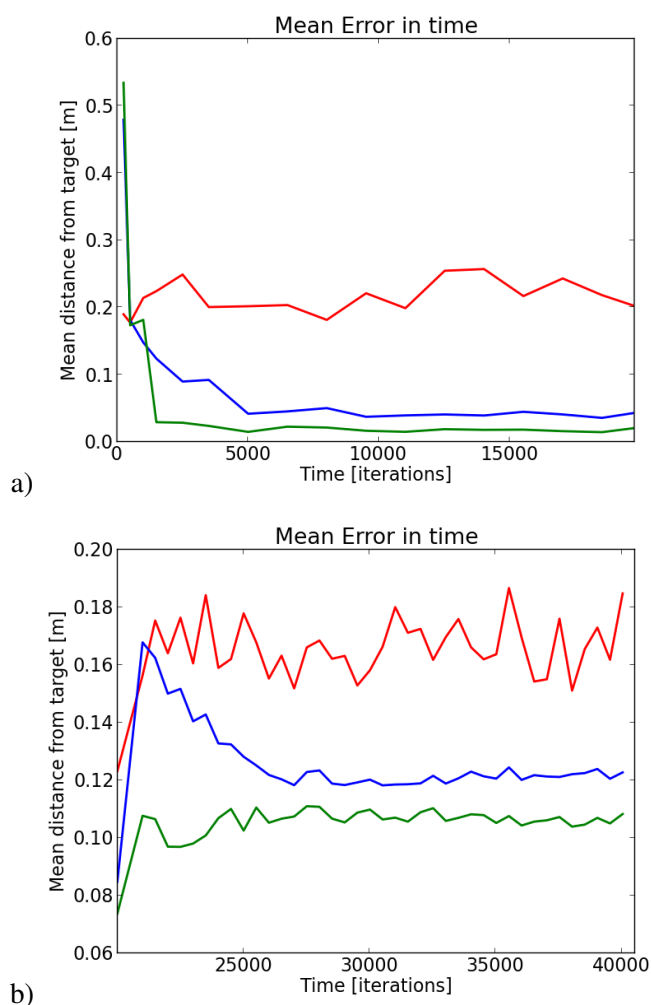


FIGURE 3.6 – Résultats en rouge pour RLS, bleu pour AS et vert pour RS (SLE) : a) Erreur moyenne lors de la tâche d’atteinte, effectuée au cours de l’apprentissage avec babillage moteur. b) Erreur moyenne lors de la tâche d’atteinte, effectuée après l’apprentissage, tandis que la géométrie du robot a changé (ajout d’un outil)

Avec la modification de la géométrie, l’évolution des performances de la tâche d’atteinte dans le temps montre que le modèle AS a besoin d’à peu près autant de temps, pour s’adapter à la présence de l’outil, que pour son apprentissage initial lors de la première phase de babillage. Ceci s’explique par le fait que ce modèle doit, pour s’adapter à l’outil, apprendre de nouveau toutes ses associations, tandis que le modèle RS n’a lui rien de nouveau à apprendre pour être efficace dans cette tâche.

De fait, la principale limitation du modèle RS que nous avons présenté, le SLE, ne tient pas à ses performances pour une tâche d’atteinte mais au nombre de neurones recrutés. En effet, le SLE doit encoder une nouvelle configuration articulaire chaque fois que la prédiction n’est pas

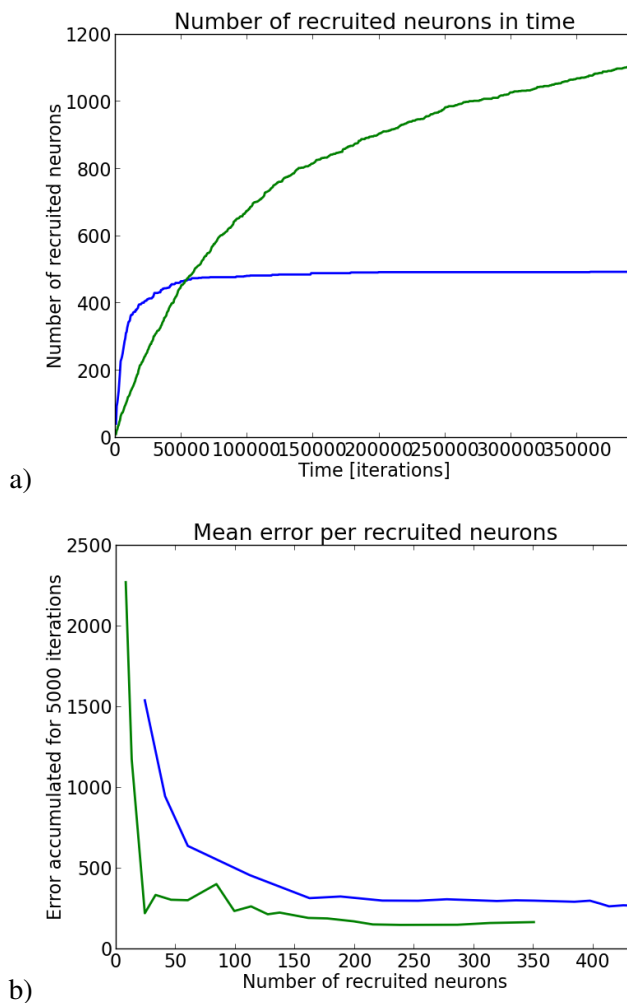


FIGURE 3.7 – Résultats en bleu pour AS et vert pour RS (avec le SLE) a) Nombre de neurones recrutés au cours du babillage. b) Performances dans la tâche d’atteinte en fonction du nombre de neurones recrutés

assez bonne, pour un seuil donné. Or l’effecteur du bras katana, ayant 4-DOF, a déjà une quantité énorme de telles lois, distinctes, de co-variations sensorimotrices dans l’espace 3D. En effet, le modèle direct du bras est basé sur de nombreuses sinusoïdes (voir sec. 4.3.2 pour voir les équations exactes), et toutes ces non linéarités sont autant de catégorisations à effectuer, pour le SLE qui repose sur un encodage des linéarités locales. Ainsi, afin d’avoir de bonnes prédictions, le SLE doit recruter beaucoup de neurones. De plus, le temps d’exploration de chaque mouvement, dans n’importe quelle configuration motrice, est assez long lors d’un babillage moteur aléatoire. Dans la figure 3.7 a), nous voyons le nombre de neurones recrutés dans le temps pour les deux modèles.

Or, il est intéressant de noter que l’évolution des performances ne suit pas celle du nombre de neurones recrutés. On pouvait en effet noter que sur la figure 3.6 a), la RS converge plus rapidement vers de bons résultats que AS. De fait, très rapidement le recrutement de neurones

dans le SLE n'a plus d'impact sur les performances de la tâche d'atteintes. Dans la figure 3.7 b), nous voyons les performances de chaque modèle en fonction du nombre de neurones recrutés. Nous pouvons voir que le modèle utilisant le SLE a besoin de moins de neurones que AS pour être précis dans la tâche.

Pour finir, notons que, sans surprise, le nombre de neurones recruté par articulation est de moins en moins important à mesure que l'on passe des articulations proximale à distale. À titre d'exemple, on note pour 470 neurones recrutés : 228, 107, 101 et 36 neurones dans l'ordre proximo-distal, et pour 364 neurones recrutés : 186, 107, 49 et 22 neurones, dans le même ordre. Ceci s'explique par le fait que les articulations proximales ont des lois potentiellement très différentes selon la configuration des articulations distales, d'autant que leur amplitude, et leur effets dans l'espace visuel sont plus importants.

3.1.5 Discussion et conclusion

Dans cette partie, nous avons étudié comment, à l'aide d'une stratégie absolue et relative, les informations sensorimotrices sont encodées. Les expériences d'atteinte dans l'espace 3D, avec les 3 simulations du bras robotique Katana, pour les 3 stratégies (AS, RS et RLS) indiquent une bonne robustesse lors de l'ajout d'un outil pour le RS. L'AS peut, elle s'adapter à l'outil, mais au prix d'un réapprentissage complet de ses associations sensorimotrices. En accord avec les conclusions du Bullock et collègues [Bullock et al., 1993], la rapidité de cette adaptation tend à rendre la RS plus pertinente pour l'utilisation d'outils, bien que les 3 peuvent coexister à différentes échelles de temps d'apprentissage.

Complémentarité des stratégies

La stratégie de renforcement (RLS), avec des performances constantes, semble une bonne solution de secours en cas de défaillance d'autres modèles. Il est ainsi possible que les 3 stratégies soient présentes dans le développement, car elles ont des propriétés différentes qui peuvent être complémentaires. En effet, les modèles AS et RS peuvent apprendre en parallèle, ils auraient alors des rôles complémentaires. Le RLS peut constituer une solution de secours en d'échec des autres stratégies. Comme nous l'avons vu, une RS est efficace pour l'utilisation d'outils. L'AS peut, elle, réaliser une tâche qui n'est pas immédiatement réalisable par la RS : par exemple, si pour atteindre une cible, la direction à emprunter depuis la position courante nécessite un retour en arrière, le RS, tel que présenter dans cette version du SLE, ne pourra faire ce mouvement. Car cela impliquerait une augmentation de la distance à la cible pendant un certain temps. Or l'AS, ne se souciant pas des étapes intermédiaires pour atteindre une configuration articulaire, et de ses conséquences sur le bras, sera, elle, capable de s'éloigner temporairement de la cible, si cela est nécessaire, pour atteindre la cible. Notons que cette problématique, qui rejoint celle de la possibilité de faire un détour pour contourner un obstacle, a été traitée par Srinivasa et collègues [Srinivasa et al., 2012]. À l'aide de mécanismes plus élaborés il est ainsi possible d'obtenir ce type de comportements (évitement d'obstacle) à l'aide d'une stratégie relative proche du SLE. Toutefois, de tels mécanismes, impliquant des répétitions mentales ("mental rehearsal"), semblent computationnellement plus lourds, quand l'AS propose en ces cas une solution immédiate.

Explosion combinatoire

Comme nous l'avons vu sur les figures 3.7 a) et b), le nombre de catégories (traduit par le nombre de neurones recrutés) est très important pour le RS. Or, la plupart de ces catégorisations n'ont pas, ou presque pas, d'impact sur les performances.

Dans le modèle actuel, lors du recrutement d'un neurone (c'est à dire d'une catégorie, ou encore d'un contexte selon le degré d'observation) l'ensemble des entrées sensorielles sont recrutés. Ceci conduit à une explosion combinatoire du nombre de neurones recrutés dans le SAW. En effet, en particulier avec le modèle RS, les co-variations de capteurs et de moteurs peuvent conduire à un nombre de combinaisons très importantes pour encoder l'ensemble des lois possible, en n'utilisant "que" la configuration proprioceptive (et non l'ensemble des senseurs, comme suggéré ci-dessus). Notons que dans [Srinivasa and Grossberg, 2008], les auteurs, qui utilisent un modèle de type RS, proche du SLE, recrutent pour leur part 77175 contextes pour encoder correctement les co-variations de leur robot à 7-DOF.

Bien que nous proposons une loi de recrutement différente, non basée sur des distances dans l'espace d'entrée, comme c'est notamment le cas dans le réseau de neurones ART (Adaptive Resonance Theory, voir [Carpenter and Grossberg, 2010]) qu'ils utilisent, mais basée uniquement sur une erreur de prédiction (voir la fig. 3.1), nous encodons encore trop de neurones. Comme nous l'avons déjà souligné, notons encore que puisque ces neurones codent pour des lois linéaires, et qu'ils sont recrutés lorsque celle-ci échoue, il serait plus judicieux que ces catégories correspondent plutôt aux frontières d'activations de ces lois linéaires. Cette idée, permettant notamment de réduire considérablement le nombre de catégories recrutées, sera détaillée au chapitre suivant 4.

Perception de l'outil, et utilisation "à dessein"

Dans nos expériences, nous avons vu comment les différentes stratégies peuvent apprendre et s'adapter à la présence d'un outil. Toutefois, la question de savoir pourquoi les outils sont si longs à utiliser à dessein (voir [Rat-Fischer et al., 2012]) reste en suspens. Si les deux modèles peuvent s'adapter, d'où vient la difficulté à *exploiter* un outil ? Un aspect de la réponse repose certainement dans la manière dont l'outil est perçu par le robot. Dans nos expériences, la question du mécanisme par lequel le bras semble être "étendu" à l'outil n'a pas été directement traitée, bien qu'il s'agisse d'une question critique : deux idées peuvent toutefois permettre de fournir des éléments de réponse.

Apprentissage de ce qu'est la "main", et transformation de cet apprentissage Une première idée importante repose sur la question de savoir comment, du point de vue d'un robot, un ensemble de données sensorielles (visuelles en l'occurrence) peut être considéré comme appartenant à soi, en premier lieu. Cette question a été abordée dans la section 2.3.3, dans laquelle nous avons proposé un modèle permettant d'associer un ensemble de points d'intérêts visuels aux flots optiques générés par les activités motrices de la main robotique. Et donc, que le robot interprète comme appartenant à soi, ce qu'il est capable de faire bouger. Nous pouvons dès lors imaginer qu'un tel mécanisme sensorimoteur, de découverte de soi, peut de la même manière s'appliquer à l'outil. Une transformation, permettant de voir la main à l'emplacement de l'outil, pourra alors être apprise. Si on utilise une AS, la transformation pourra être apprise par le mécanisme basé sur les neurones à champs de gain, tel que montré dans la section 2.2. Si l'on utilise

une RS, comme le SLE, alors encoder des contextes relatifs à la vision de la main, et ce grâce à l'activité motrice de saisie de l'outil, permettrait d'apprendre cette transformation *au même titre* que les autres lois. L'intérêt d'une telle manière de faire réside dans le fait qu'alors l'outil serait appris *de la même manière* que le reste du corps du robot. Un même mécanisme serait alors responsable de l'apprentissage des lois sensorimotrices relatif tant au robot, qu'au monde extérieur. Ceci est d'ailleurs cohérent avec les données neurobiologiques et psychologiques (voir sec. 1.1.1), et permet, en outre, de ne pas nécessiter de choix *a priori* du modélisateur sur ce que le robot est ou n'est pas : ce qui se traduit généralement par un choix d'entrées spécifiques (comme par exemple la seule proprioception), ou encore l'ajout de différents mécanismes d'apprentissage (comme une AS et des neurones à champs de gain, par exemple). Ce sera l'objet de la section qui suit (sec. 3.2).

Utilisation “à dessein” de cette transformation La seconde question que l'on peut se poser concerne donc l'utilisation qui est faite de ces différents apprentissages. Par utilisation “à dessein” nous entendons ici distinguer le comportement qui nous intéresse de ceux relevant du hasard, de l'habitude, ou d'un renforcement provenant d'une récompense. Il s'agit plus de questionner comment une telle transformation peut devenir un outil permettant de réaliser un but donné. Bien que les mécanismes de hasard, d'habitude, ou de renforcement aient pu contribuer à la formation de cette connaissance, on peut se demander s'il est possible de concevoir la saisie d'un outil comme étant une sorte de sous-but. Mais un sous-but d'un type particulier, qui pourrait, par exemple, ne pas faire intrinsèquement partie ni de la tâche telle qu'elle est définie, ou encore ne pas faire partie de la manière habituelle de résoudre cette tâche.

La question de la capacité à utiliser “à dessein” un outil recoupe celle d'être à même d'utiliser, à dessein, différents apprentissages sensorimoteurs c'est-à-dire des lois sensorimotrices. Et ainsi, d'être à même de “jouer” avec les lois de transformations pouvant encoder tant son propre corps, que des outils, comme nous l'avons développé dans le précédent paragraphe.

Or un ensemble de problèmes se pose alors : il faut d'une part qu'un apprentissage n'efface pas un précédent apprentissage, si l'on veut pouvoir utiliser l'un ou l'autre selon les cas. Or c'est ce qui se passe dans notre simple implémentation de l'AS, et c'est également le cas lors du simple ajout d'un neurone avec RS (voir le chapitre 4 et la question de “l'oubli catastrophique”). Et, d'autre part, il faut que chaque apprentissage soit “mobilisable”, afin de pouvoir être utilisé si nécessaire. Or ce qui doit être mobilisé, pour faire appel à une loi sensorimotrice donnée, c'est ce qu'on appelle un contexte, dans la mesure où celui-ci détermine la loi à utiliser. La difficulté repose alors tant sur la capacité de pouvoir mobiliser un contexte lorsque celui-ci pourrait être utile, que dans les définitions possible de ce contexte. Dans le cas du SLE, le contexte, ce que nous avons appelé catégorie, se traduit par un neurone. Ce contexte peut potentiellement représenter plusieurs choses. Si l'on ajoute des entrées visuelles issues de traitements, comme dans la section 2.3.3, alors il est important de noter que “avoir un outil dans la main” peut définir un contexte, et ce exactement au même titre que le type de contexte recruté dans la figure 3.4 : or un des contextes concerne ici l'encodage d'une non-linéarité dans une loi visuo-motrice (ce qui est relativement bas niveau), quand l'autre concerne la présence d'un outil dans la main (ce qui peut sembler relativement haut niveau). En dépit de l'écart conceptuel que l'on perçoit, en tant qu'observateur extérieur, entre les différents niveaux de traitements ou dans la complexité qui est encodée par ces contextes, nous pouvons noter qu'il s'agit chaque fois de connaissances qui peuvent potentiellement permettre de résoudre une tâche donnée. Notons également que la

complexité de ce qui est encodé dans un contexte ne présume en aucune façon de la complexité de la tâche qui pourra être résolue grâce à la mobilisation de ce contexte. Notons pour finir que, dès lors, ce qui définit pour un robot doté d'un tel modèle la perception de son propre corps, se comprend bien plus en termes de contextes "*mobilisables*" *en permanence* au sein de ses lois sensorimotrices (un robot peut toujours plier son bras et être dans ce contexte) que comme un ensemble pré-déterminable de senseurs et d'actionneurs. Seul un babillage non contraint (autrement que par un relâchement progressive de contraintes développementales), et la variabilité de tout ce qui ne constitue pas son propre corps, fera statistiquement aboutir la perception d'un robot de son propre corps à son "véritable" corps. Cette utilisation "à dessein" de lois sensorimotrices sera exploitée dans la section qui suit.

3.2 Séquences d'actions, basées sur un modèle relatif et des simulations

3.2.1 Introduction

Continuité avec les précédents modèles

Nous avons vu dans la partie précédente un modèle issue d'une stratégie relative, le SLE. Celui-ci permet d'éviter de tout réapprendre lorsqu'un outil est ajouté. Dans la lignée des conclusions de la section 2.3, ce modèle suppose un apprentissage différent par senseurs, et apprend les contextes relatifs à ses lois sensorimotrices : une erreur se trouve corrigée par un contexte dédié.

De plus, contrairement au modèle de neurones à champs de gain, la manière dont chaque modalité influence une autre modalité n'a pas à être décidée par le modélisateur (séquence vision puis proprioception par exemple, voir la figure 2.8), dans la mesure où potentiellement chaque entrée sensorielle peut influencer chaque couple senseur-moteur (dans le chapitre 4 nous verrons une manière de sélectionner les senseurs pertinents parmi eux). Et de la même manière qu'avec le modèle de neurones à champs de gain, c'est, avec le SLE, une transformation qui est apprise, basée sur les moteurs, où chaque variation apprise est une transformation dans l'espace du senseur concerné.

Enfin, les lois de covariations entre senseurs et moteurs sont elles-mêmes variables, en fonction du contexte. Le contexte co-varie avec la loi sensorimotrice qui s'applique. Et c'est ce principe qui sera utilisé faire usage des contextes "à dessein", c'est-à-dire en vue d'un but, et résoudre une tâche d'une manière que le robot n'a jamais expérimenté auparavant, en faisant appel à une loi, et donc à un contexte.

Lien avec des théories psychologiques, et avec des problématiques de l'"open-ended"

Nous allons présenter dans cette partie un modèle computationnel permettant d'obtenir des séquences d'actions, basées sur le principe idéomoteur (voir [James, 1890], et la section 1.1.3).

Il est inspiré des modèles théoriques développés par Pezzulo et Castelfranchi [Pezzulo and Castelfranchi, 2009]. Pour ces auteurs, les comportements guidés par un but demandent d'aller "au-delà" d'une réaction immédiate à l'environnement, c'est-à-dire de la simple association de perceptions aux actions. Pezzulo et Castelfranchi formulent des modèles théoriques basés sur la prédiction des conséquences de l'action, en insistant sur la relation étroite entre l'imagerie mentale et le système moteur. Ils visent à aller au-delà de l'apprentissage des modèles directs et inverses, en tant que mécanismes distincts, et suggèrent un apprentissage bidirectionnel des actions et des effets.

Le modèle que nous allons présenter est dans la continuité du SLE, qu'il utilise, et s'appelle Modèle Sensorimoteur Dynamique (DSM). En plus des travaux de Pezzulo et Castelfranchi, ses différents aspects sont inspirés des théories que nous avons développées dans la section 1.1.

Ce modèle computationnel prend en entrée les données brutes des senseurs et des moteurs, et fournit :

- un apprentissage incrémental de la variation des senseurs en fonction des moteurs (soit le modèle direct du SLE).

- un mécanisme “open-ended” qui utilise le modèle direct pour résoudre à la volée des tâches d’une manière qui n’a pas été expérimentée. Le modèle proposé pourra acquérir des compétences de bas et de haut niveau grâce à des mécanismes d’apprentissages identiques.

DSM utilise l’encodeur de loi sensorimotrice (SLE, voir section 3.1.1), d’une manière proche de celle présentée dans la section 3.1.4. Toutefois le mécanisme a été généralisé, et sera développé dans la section suivante 3.2.2. Nous montrerons enfin le modèle DSM dans son ensemble dans la partie 3.2.3. Enfin, nous montrerons des expériences utilisant ce modèle sur robot réel dans la section 3.2.4.

3.2.2 Le Simulateur de Lois Sensorimotrices (SLS)

Le SLE (voir section 3.1.1) est le seul mécanisme d’apprentissage dans DSM. Ainsi dans DSM ne sont apprises que les lois prédictives sensorimotrices de covariations entre les senseurs et les moteurs. Mais ce mécanisme prédictif sera ensuite utilisé pour l’action [Flanagan et al., 2003] par un mécanisme indépendant du processus d’apprentissage. Notons qu’étant donné qu’aucun modèle inverse n’est appris, nous évitons le problème lié à l’apprentissage d’une solution unique parmi une infinité (voir section 1.2.1), ainsi que celui de l’apprentissage de multiples modèles inverses, qui ne sont utiles que dans certains contextes et qui sont difficiles à apprendre et à sélectionner (voir [Wolpert and Kawato, 1998] et section 1.2.1).

Dans DSM, l’action est choisie sur la base d’un mécanisme d’affordance. L’idée est que l’inversion des associations apprises entre action et effet (laquelle est théoriquement possible, selon le principe idéomoteur) déclenchera des possibilités d’action, que nous choisissons d’appeler des affordances. Ici les affordances sont considérées comme un désir d’effectuer les variations sensorielles possibles, lesquelles sont données en sortie du SLE lorsqu’on lui fournit des entrées simulées. De la sorte, la seule observation du monde extérieur déclenche des affordances, qui sont ensuite utilisées pour effectuer des actions orientées vers un but. Et ces buts sont eux-mêmes perçus en tant qu’ensemble de variations sensorielles désirées, depuis la position courante à celle du but.

En accord avec Moller et collègues [Möller and Schenck, 2008], nous proposons que les affordances représentent les prédictions calculées par le SLE lorsque les entrées motrices \mathbf{M} ne sont plus une copie des commandes motrices réelles \mathbf{M}^{Real} , mais proviennent d’une simulation \mathbf{M}^{Sim} . Comme le SLE apprend la loi linéaire locale entre la variation du senseur prédit et les entrées motrices, étant donné \mathbf{M}^{Sim} il est possible de calculer un ensemble de magnitudes motrices requis pour atteindre un but, au sein du contexte local.

Nous proposons d’aller plus loin, et d’utiliser également un mécanisme de simulation avec la seconde entrée du SLE, c’est-à-dire en donnant en entrée \mathbf{S}^{Sim} . Le résultat de la prédiction n’est alors pas une affordance immédiate - où une action dans le contexte existant pourrait déclencher le résultat prédit - mais plutôt une sorte d’“affordance contextuelle”, ou encore une affordance différée. Autrement dit, il s’agit d’une affordance *d’un contexte* dans lequel une affordance immédiate serait possible. En d’autres mots, une prédiction basée sur \mathbf{S}^{Sim} peut être utilisée pour déclencher des sous-buts \mathbf{S}^{Des} à atteindre, lesquels, une fois achevés, permettent au robot d’être dans un contexte dans lequel certaines actions utiles sont possibles.

Au lieu d’apprendre un modèle inverse, nous proposons ainsi un Simulateur de Loi Sensorimotrice (SLS), dans lequel plusieurs mécanismes sont nécessaires, représentés dans la figure 3.8. La plupart de ces mécanismes sont similaires à ceux décrits par Pezzulo et Castel-

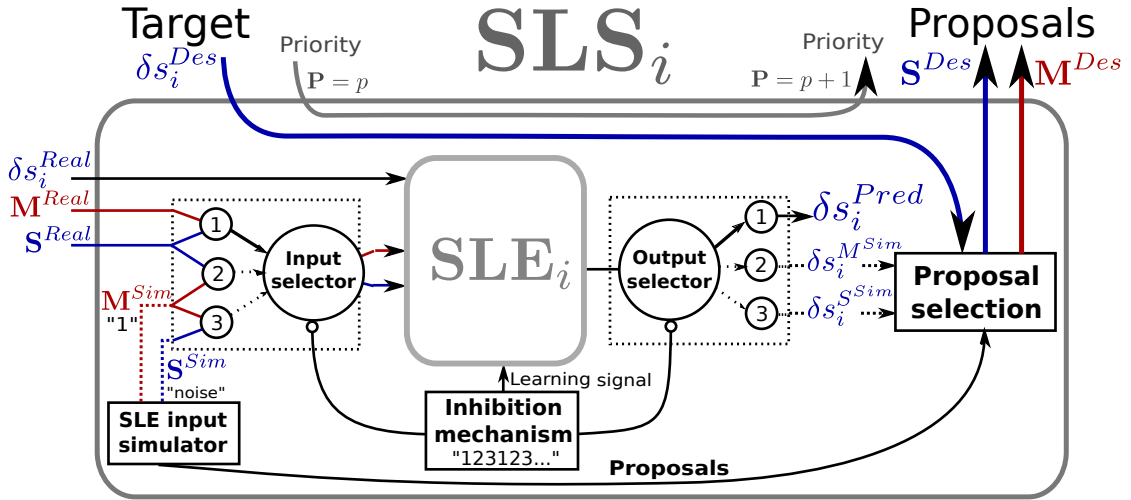


FIGURE 3.8 – Simulateur de Lois Sensorimotrices (SLS) pour le capteur i . Un mécanisme d'inhibition permet de fournir en entrée du SLE trois entrées différentes. Celles-ci proviennent soit des données réelles, soit de données motrices simulées et des données sensorielles réelles, soit de données motrices et sensorielles simulées. Pour un but donné, le SLS calcule les sous-buts souhaités \mathbf{S} ainsi que les commandes motrices \mathbf{M} , en fonction des résultats fournis par les différentes simulations (notons qu'aucun apprentissage n'est ici impliqué).

franchi dans [Pezzulo and Castelfranchi, 2009] (et correspondent plus spécifiquement au stade 2 décrit dans cet article). Premièrement, nous proposons d'utiliser un mécanisme d'inhibition pour utiliser le SLE avec d'autres vecteurs d'entrées. Les mécanismes d'inhibition sont fréquents dans le cerveau, et Pezzulo et Castelfranchi suggèrent, dans le mécanisme décrit dans leur article, que l'inhibition pourrait agir comme un commutateur. Ainsi dans ce papier le mécanisme d'inhibition ne sera pas développé, et nous l'utiliserons comme un simple commutateur permettant de varier entre 3 combinaisons différentes d'entrées. Comme mentionné dans la section 3.1.1, les entrées à considérer pour effectuer des prédictions sont les vecteurs de capteurs \mathbf{S} , et de moteurs \mathbf{M} .

Nous proposons ensuite un mécanisme permettant de proposer de telles simulations, notées \mathbf{S}^{Sim} et \mathbf{M}^{Sim} . De par la linéarité de chaque apprentissage, des valeurs non nulles (en l'occurrence 1) sont suffisantes pour tous les éléments du vecteur \mathbf{M}^{Sim} . Le vecteur \mathbf{S}^{Sim} , quant à lui, est arbitrairement rempli par des valeurs aléatoires. Ainsi, le mécanisme d'inhibition du SLS varie entre les 3 combinaisons d'entrée suivantes (également représentées dans la figure 3.8) :

- ① en utilisant les entrées réelles \mathbf{M}^{Real} and \mathbf{S}^{Real} . Cette entrée est utilisée pour l'apprentissage. La sortie δs_i^{Pred} est la variation sensorielle prédite du capteur i .
- ② en utilisant les entrées sensorielles réelles \mathbf{S}^{Real} mais les entrées motrices simulées \mathbf{M}^{Sim} . La sortie $\delta s_i^{M^{Sim}}$ est la valeur d'"affordance", et correspond à la variation sensorielle prédite du capteur i dans le cas où le robot exécuterait la commande motrice \mathbf{M}^{Sim} . Notons qu'une valeur élevée de $\delta s_i^{M^{Sim}}$ indique une grande efficacité du moteur correspondant, pour le déplacement du capteur i dans le contexte existant. Cette valeur sera plus tard utilisée pour le calcul d'une valeur d'efficacité E .

- ③ en utilisant les entrées simulées \mathbf{M}^{Sim} et \mathbf{S}^{Sim} . La sortie $\delta s_i^{S^{Sim}}$ reflète le caractère approprié du contexte simulé (une “affordance de contexte”), autrement dit un "contexte" dans lequel les commandes motrices \mathbf{M}^{Sim} déclenchent une variation désirée du senseur i .

Considérant les 2 sorties simulées données par ② et ③, le mécanisme de sélection de proposition (“Proposal Selection”) détermine les \mathbf{M}^{Sim} et \mathbf{S}^{Sim} simulés qui sont pertinents pour un δs_i^{Des} désiré. Il s’agit d’une sorte de concurrence entre affordances, mais ici également entre affordances immédiates et différées (affordances de contextes). Dans cette implémentation, seul un signal désiré est envoyé, \mathbf{M}^{Des} ou \mathbf{S}^{Des} . Dans le premier cas, si l’affordance immédiate utilisant \mathbf{M}^{Sim} est assez forte, alors une commande motrice \mathbf{M}^{Des} est désirée. Autrement, dans le second cas si l’affordance différée utilisant \mathbf{S}^{Sim} est assez forte, un sous-but \mathbf{S}^{Des} est désiré. Un seuil Th_2 est utilisé pour déterminer si une affordance est suffisamment forte. Ainsi, si ce seuil est trop bas les commandes motrices seront presque toujours désirée. Inversement, si ce seuil est trop élevé aucune des commandes motrices ni aucun des sous-buts ne seront jamais désirés.

Comme nous l’avons dit, dans le premier cas si $\delta s_i^{M^{Sim}}$ est au-delà du seuil Th_2 (c’est-à-dire si l’efficacité E est suffisamment élevée), nous calculons et envoyons une commande motrice appropriée, en utilisant la loi linéaire locale entre les senseurs et les moteurs :

$$M^{Des} = \frac{\delta s_i^{Des}}{\delta s_i^{M^{Sim}}} \quad (3.20)$$

Cette commande motrice peut devenir imprécise, de par un apprentissage approximatif, ou de par les changements de contextes au cours de la trajectoire. Cependant, la commande M^{Des} est calculée à chaque itération et est donc continuellement adaptée.

Dans le second cas, si $\delta s_i^{S^{Sim}}$ est au-deçà du seuil Th_2 , le modèle teste l’efficacité du contexte qui donne la plus grande affordance différée. Pour ce faire, à chaque itération le mécanisme de sélection vérifie si $\delta s_i^{S^{Sim}_{Max}}$ a la valeur la plus élevée jamais rencontrée. Si la valeur courante est plus élevée, il actualise le couple formé par $\{\mathbf{S}^{Sim}_{Max}, \delta s_i^{S^{Sim}_{Max}}\}$ avec les valeurs courantes $\{\mathbf{S}^{Sim}, \delta s_i^{S^{Sim}}\}$. Ensuite, si $\delta s_i^{S^{Sim}_{Max}}$ est au-delà du seuil Th_2 , l’entrée simulée correspondante est envoyée : $\mathbf{S}^{Des} = \mathbf{S}^{Sim}_{Max}$. Le diagramme dédié au SLS, figure 3.10, illustre ces mécanismes internes.

3.2.3 Architecture du modèle sensorimoteur dynamique (DSM)

Le Modèle Sensorimotrice Dynamique (DSM) traite différents buts désirés, et commandes motrices désirées. L’idée est que, pour un but donné, DSM utilise les différents SLS pour trouver les commandes motrices et les sous-buts les plus appropriés pour atteindre ce but, en le faisant agir, et si nécessaire exécuter des séquences d’actions jamais apprises.

Un but est un vecteur \mathbf{S}^{Des} de l senseurs, avec l plus petit ou égal au nombre total de senseurs du robot. La distance entre les valeurs courantes des senseurs \mathbf{S}^{Real} , et \mathbf{S}^{Des} , qui est donc la distance entre le contexte courant et le contexte désiré, est calculée pour obtenir l différents δs_i^{Des} , lesquels sont envoyés aux SLS_i correspondants, voir la figure 3.9.

Ces derniers envoient des sous-buts \mathbf{S}^{Des} ou des commandes motrices \mathbf{M}^{Des} . Un mécanisme de sélection est donc nécessaire ici pour traiter les buts et les sous-buts récursifs, ainsi que les différentes commandes motrices, éventuellement en opposition sur les mêmes moteurs. Dans

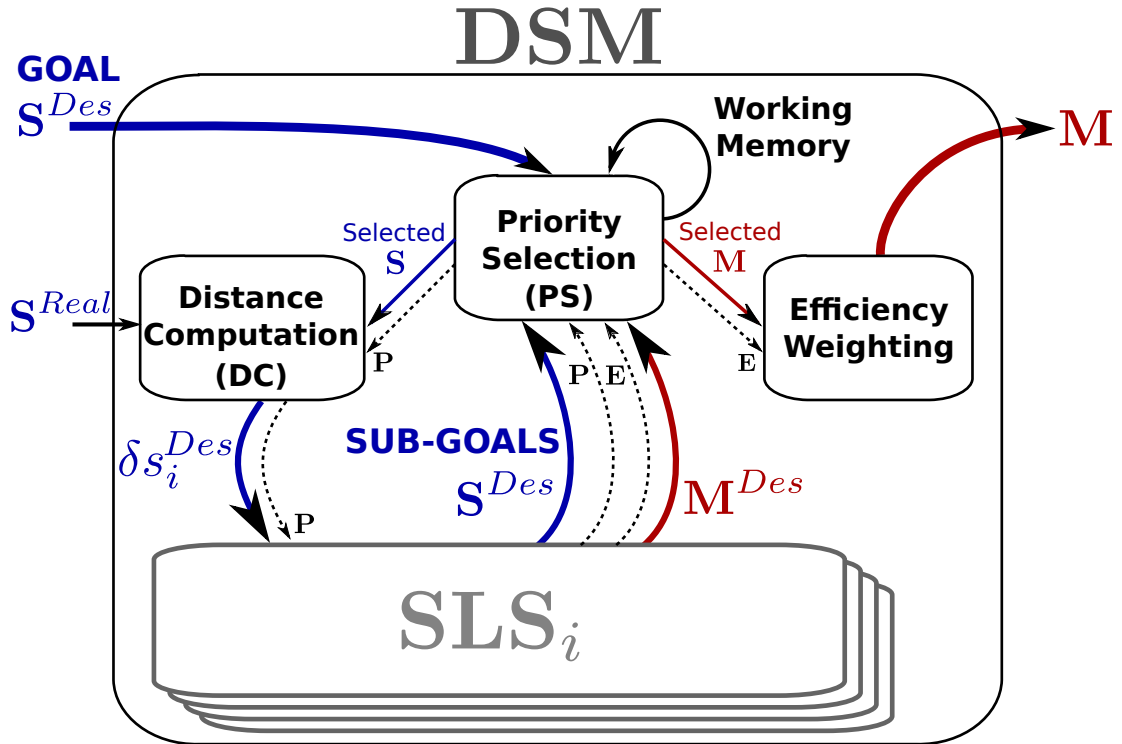


FIGURE 3.9 – Le modèle Sensorimoteur Dynamique (DSM). Pour un but initial donné, DSM génère des sous-buts et sélectionne les commandes motrices les plus efficaces, provenant de différents SLS. Il utilise un mécanisme de sélection (représenté comme le “Priority Mechanism”, PS) pour choisir entre les sous-buts et le but initial, si ce dernier est immédiatement réalisable. Il envoie ensuite, pour chaque senseur, aux différents SLSs correspondants, la variation sensorielle désirée qui permettra d’atteindre le but sélectionné. Une mémoire de travail est utilisée pour conserver en mémoire le but sélectionné.

cette implémentation, nous proposons un mécanisme de priorité, où chaque but (ou sous-but, ils sont traités de la même manière) est couplé à une valeur de priorité P . Pour le but initial, nous fixons $P^{Goal} = 1$. Le mécanisme sélectionne simplement la priorité avec la valeur maximale, et envoie les S^{Des} et M^{Des} sélectionnés. Le S^{Des} , couplé à sa priorité $P = p$, est alors envoyé au mécanisme de calcul de distance (DC), lequel calcule l’ensemble des distances au vecteur sensations réelles courantes. Les couples distances - priorités sont alors envoyés aux SLSs correspondants. Ceux-ci, en retour, envoient éventuellement de nouveau S^{Des} ou M^{Des} , couplés cette fois à la priorité $P = p + 1$.

Une fois que le mécanisme de sélection a opéré, il est encore possible, pour certaine commandes motrices, d’avoir des valeurs opposées sur les mêmes moteurs (chaque SLS peut envoyer un vecteur M^{Des} concernant tous les moteurs, et ayant la même priorité). Le problème avait été abordé dans la section 3.1.4. Suivant un principe identique pour le résoudre, nous avons donc ajouté, à la suite du mécanisme de sélection, un second mécanisme de pondération par l’efficacité (“Efficiency Weighting”), qui concerne l’ensemble des M^{Des} en sortie du mécanisme de sélection. Les commandes motrices provenant du SLS sont individuellement couplées avec une valeur d’efficacité E (voir le point ② de la section 3.2.2). En ce qui concerne la manière de calculer cette valeur E , on pourrait défendre l’idée que la valeur $\delta s_i^{M^{Sim}}$ correspondant à ce que

nous avons appelé la valeur d'affordance, mesurant l'importance de la variation prédite d'un senseur en cas d'activité motrice, serait suffisante pour remplir ce rôle : nous aurions alors une simple compétition entre affordances. Cependant, cette valeur ne prend pas en compte "l'utilité" de cette affordance dans le cadre du but poursuivi, autrement dit en quoi celle-ci est utile à l'accomplissement de la tâche en cours. Nous proposons donc plutôt que $E = \delta s_i^{M^{Sim}} * \delta s_i^{Des}$, de sorte que l'efficacité correspond finalement à la potentielle quantité de travail nécessaire pour atteindre le but courant. Bien qu'il existe potentiellement de multiples manières de calculer la commande motrice M qui sera réellement envoyée au robot, en se basant sur l'efficacité E (comme les solutions basées sur une combinaison linéaire de toutes les commandes reçues, qui auront pour défaut de rendre imprédictible la commande motrice réellement effectuée), la commande motrice qui sera envoyée en sortie de EW sera, pour chaque moteur, la commande dont la valeur d'efficacité est maximale. Notons que comme E reste fortement lié à la valeur affordancielle au sein du contexte courant, cette méthode sélectionnant l'efficacité maximale reste très proche de l'idée de la compétition entre affordance (voir [Cisek, 2007] et la section 1.1.5).

Dans la figure 3.10, on peut voir une implémentation de DSM dans un diagramme qui inclut les équations principales des mécanismes internes. Par souci de simplicité, celui-ci est simplifié et n'inclut pas le mécanisme de pondération par l'efficacité.

Les buts, c'est-à-dire des paires constituées d'un vecteur contenant une configuration de senseurs désirée, et d'une valeur de priorité, sont envoyés au module de sélection de priorité, lequel sélectionne le but ayant la priorité la plus élevée. Le module de calcul de distance envoie ensuite chaque variation de senseur requise, associée à sa priorité, au SLS correspondant. Chaque SLS utilise alors des simulations, sur le SLE auquel il est lié, pour calculer la variation du senseur dans le cas d'une activité motrice dans le cadre du contexte en cours, ou encore dans le cadre d'un autre contexte. En fonction d'un seuil, les SLSs envoient soit une commande motrice, soit un contexte désiré (c'est-à-dire une configuration de senseurs) en tant que nouveau sous-but associé à une priorité plus élevée.

DSM permet aux commandes motrices d'être exécutées dès que possible, tandis que des comportements séquentiels peuvent émerger à la volée. Prenons un exemple. Soit un but consistant à atteindre un point donné (tâche d'atteinte). Le, ou les SLSs correspondant à la main envoient des commandes motrices M^{Des} permettant de rapprocher la main robotique de la cible : si un tel mouvement est possible, ils n'envoient donc pas de S^{Des} à ce stade. Or si la main ne peut pas, ou plus se rapprocher suffisamment de la cible, alors le SLS concerné peut envoyer un contexte S^{Des} permettant de rapprocher malgré tout la main, si un tel contexte existe. En supposant qu'avoir un outil dans la main ait été appris comme permettant une extension de la longueur du bras, alors un S^{Des} , correspondant à avoir un sous-but "avoir l'outil dans la main", peut être envoyé. La section suivante reprend cette expérience et l'expliquera en détail. Mais notons que de par l'interaction avec les autres SLSs, le SLS de la main qui a envoyé ce sous-but, afin de pouvoir rejoindre la cible, peut se voir attribuer une nouvelle tâche qui consistera justement à réduire la distance de la main à l'outil, avec de fait un nouveau δs_i^{Des} . Mais alors nous rencontrons le problème suivant. Le SLS émetteur du sous-but se trouve lui-même attribué d'une nouvelle tâche, et ainsi le sous-but, pourtant à l'origine de cette nouvelle tâche, ne sera plus émis et donc maintenu, et se trouve en quelque sorte oublié. Ce problème trouve son origine dans le fait que, au sein de cette séquence particulière hiérarchisant les différents buts de par le mécanisme de priorité, le SLS de la main se retrouve à deux endroits hiérarchiques différents, et ne peut donc à la fois avoir le rôle d'envoyer constamment un sous-but, jusqu'à son achèvement, et celui

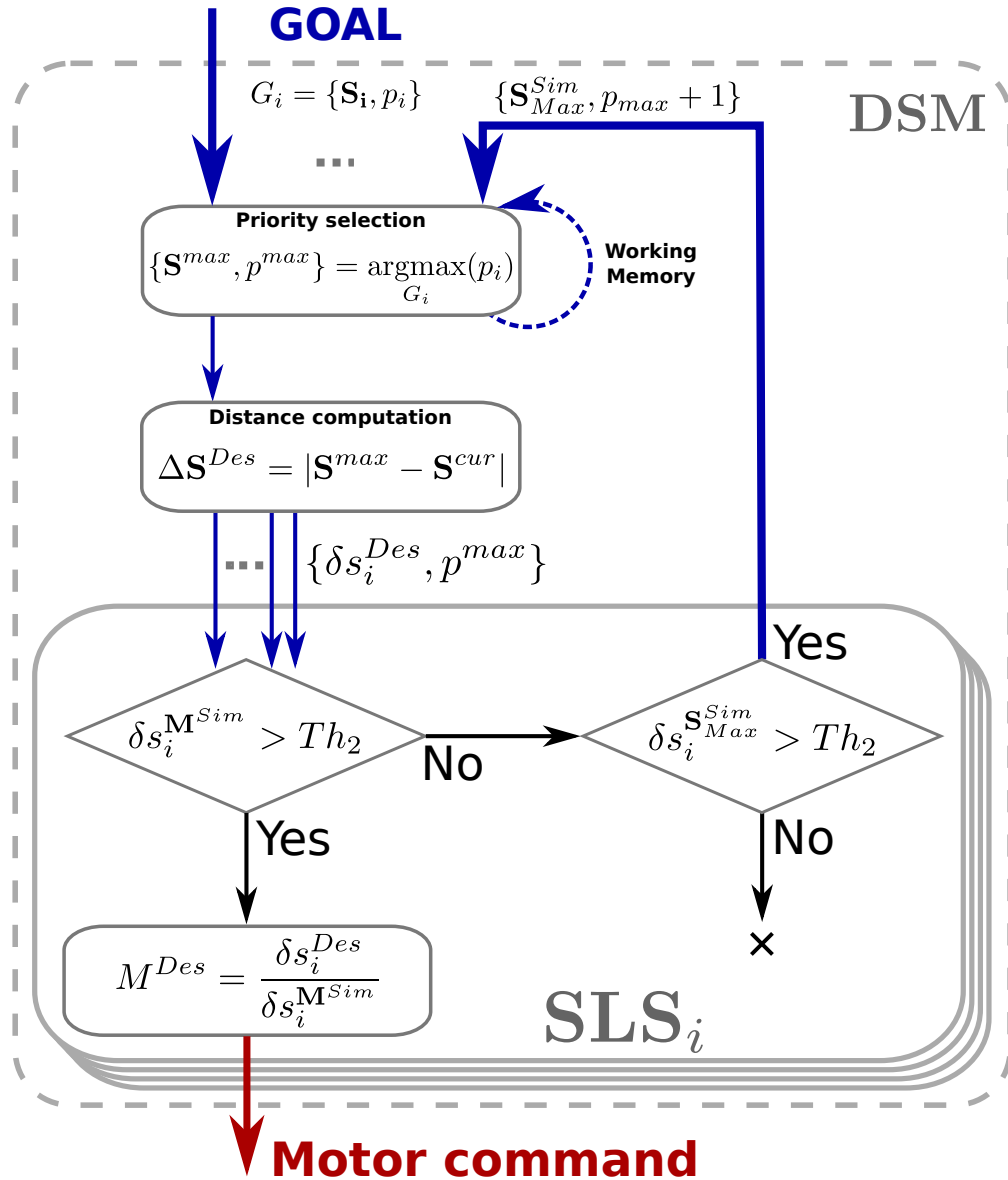


FIGURE 3.10 – Diagramme de l’implémentation de DSM utilisé dans nos travaux.

d’exécuter les tâches nécessaire à son exécution. Pezzulo et Castelfranchi suggèrent qu’une mémoire de travail est nécessaire pour maintenir des but distants (voir [Pezzulo and Castelfranchi, 2009]). Adaptant leur suggestion, tous les s_i^{Des} du vecteur \mathbf{S}^{Des} sont stockés dans une mémoire de travail (“working memory”) au sein du mécanisme de sélection de priorité, lequel est aussi responsable de leur effacement, un par un, lorsque $|s_i^{Des} - s_i^{Real}| < Th_2$.

3.2.4 Expériences sur robot réel

La capacité de faire des séquences à la volée avec DSM est testée par une expérience sur robot réel. La tâche consiste à maximiser un signal de récompense, en atteignant une cible donnée avec la main du robot. Deux outils différents sont par ailleurs disponibles, avec des formes différentes. En permettant d'étendre l'espace atteignable ils peuvent, suite à un apprentissage, donner lieu à la propriété d'étendre le schéma corporel (c'est-à-dire la géométrie du bras) dans différentes directions. Ainsi, l'outil 1 prolongera le bras verticalement et l'outil 2 le prolongera horizontalement. Afin de montrer les différentes fonctions de ces outils, une des articulations du bras est bloquée de sorte que chacun des deux outils puisse avoir son utilité propre, et puisse être utilisé dans des situations différentes. Ainsi, si les outils sont inutiles lorsque la cible est accessible, l'outil 1 devient utile lorsque la cible est hors de portée sur l'axe Y, et l'outil 2 devient lui utile lorsque la cible est hors de portée sur l'axe X. Ces règles sensorimotrices sont apprises par les différents SLEs, ce qui permet ainsi d'observer comment notre modèle est capable de maximiser la récompense en tirant partie de sa capacité à faire des séquences.

Notre installation se compose, comme précédemment, d'un bras Katana à 6 degrés de libertés dont l'un est une pince permettant la saisie, d'une caméra, de deux outils et d'une cible, voir la figure 3.11. La caméra est utilisée pour reconnaître et localiser, dans l'espace rétinien, le bras, les deux outils et la cible sur l'image. Ceci est fait grâce à un algorithme de reconnaissance de formes, basé sur des points d'intérêts locaux (voir section 2.3). Cela permet d'obtenir un vecteur de 12 neurones, 4 pour la présence des différents objets (le bras, les deux outils et la cible), et 8 pour leurs positions sur les axes X et Y de l'image.

Dans cette expérience, la cible est hors d'atteinte, on attend ainsi du robot que sa main tente d'atteindre l'outil, afin de s'en saisir. Lorsque celui-ci est effectivement saisi, le robot perçoit la position de sa main comme étendue à celle de la position de l'outil. Par ailleurs, un signal de récompense est également donné au robot : ce signal augmente lorsque la distance qui est perçue entre la main robotique et la cible $s_{H \rightarrow Target}$ diminue.

Nous avons déterminé les valeurs des deux seuils par l'expérience : dans notre implémentation, tous les senseurs varient entre 0 et 1, et nous avons choisi $Th_1 = 0.1$ et $Th_2 = 0.05$.

Nous utilisons trois SLEs. Chacun d'entre eux est dédié à la prédiction de la variation d'un senseur, et dispose de senseurs et de moteurs en entrée (voir la figure 3.1). Nous présentons ici les entrées de chaque SLE (voir figure 3.12) : notons que ces choix ont été effectués pour permettre à chaque SLE de pouvoir effectuer ses prédictions, mais nous développerons dans le chapitre suivant (chapitre 4) une extension de DSM, permettant aux SLEs de sélectionner eux-mêmes leurs entrées.

- SLE_{H_x} prédit la variation de la position perçue de la main robotique, sur l'axe X. Il prend la proprioception θ_1 du premier degré de liberté comme entrée sensorielle. Etant donné que le bras Katana a son propre PID, nous n'avons pas vraiment de contrôle direct des commandes motrices. Nous utilisons donc comme entrée motrice la variation de la proprioception $\delta\theta_1$. Un autre couple de senseur et de moteur est utilisé pour prédire la variation induite par l'outil 2 (modélisé comme une extension du schéma corporel). Il prend la distance $s_{H \rightarrow Tool2}$, entre la main et l'outil 2 dans l'espace rétinien, comme entrée sensorielle, et la commande motrice δ_{Grasp} , correspondant à la pince, permettant la saisie comme entrée motrice.

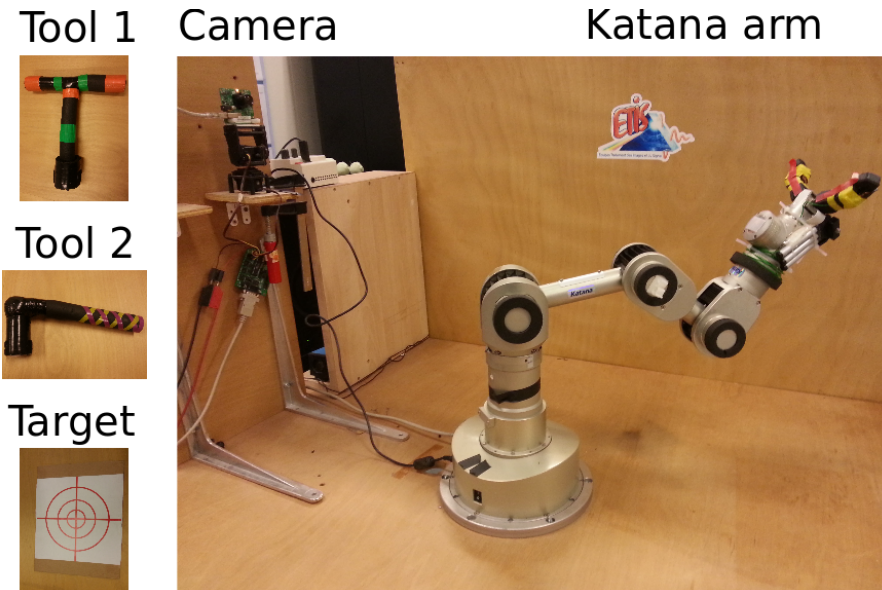


FIGURE 3.11 – Système expérimental

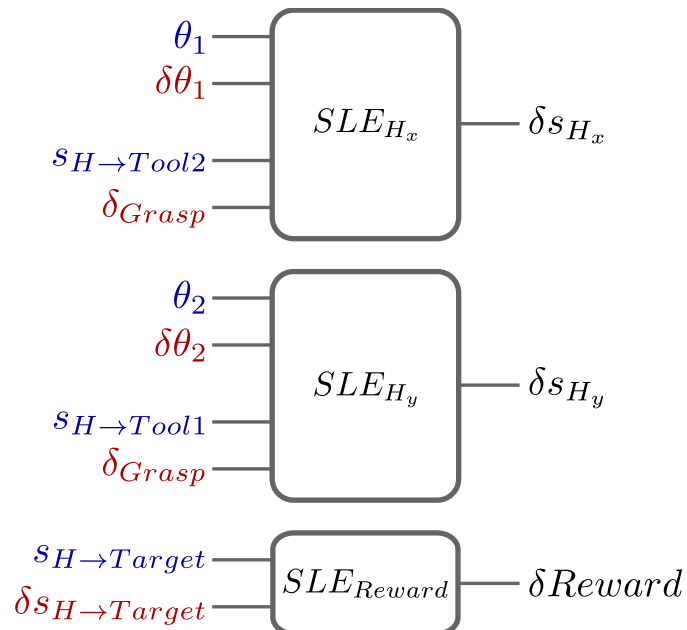


FIGURE 3.12 – Les trois SLEs utilisés pour les expériences, avec leurs entrées sensorielles (en bleu) et motrices (en rouge) à gauche, et les valeurs prédites à droite.

- SLE_{H_y} prédit la variation de la position perçue de la main robotique, sur l'axe des Y. Elle prend la proprioception θ_2 du deuxième degré de liberté comme entrée sensorielle. Pour la même raison que celle exposée ci-dessus, $\delta\theta_2$ est l'entrée motrice. Un autre couple de senseur et de moteur est également utilisé par ce SLE pour prédire la variation induite par

l'outil 1. Il prend la distance $s_{H \rightarrow Tool1}$, entre la main et l'outil, comme entrée sensorielle, et la commande motrice δ_{Grasp} permettant la saisie comme entrée motrice.

- SLE_{Reward} prédit la variation d'un signal de récompense associé à la tâche d'atteinte. En utilisant la propriété du codage commun, nous utilisons comme entrée motrice la variation de la distance perçue (visuelle) entre la main robotique, et la cible $\delta s_{H \rightarrow Target}$. Cette distance est linéaire avec la récompense, et n'est pas dépendante du contexte. De sorte qu'aucun contexte n'est nécessaire (une loi linéaire est suffisante). Néanmoins, nous avons pris $\delta s_{H \rightarrow Target}$ en entrée sensorielle.

Rappelons que dans l'ensemble de l'architecture DSM, les SLEs sont les seuls mécanismes d'apprentissage. Ils apprennent indépendamment les uns des autres, et indépendamment de toute tâche. Chaque SLE est associé à un SLS, qui lui aussi s'exécute en parallèle, et indépendamment des autres.

L'apprentissage des trois SLEs est alors réalisé, lors d'un babillage moteur couplé avec la prise des deux outils que l'expérimentateur lui aura arbitrairement mis dans la main (pince robotique). Aucune différence majeure n'est notée lorsque l'on utilise les SLE sur robot réel, en comparaison avec les simulations. Cependant les entrées visuelles bruitées (voir la position estimée de la main lorsque celle-ci effectue pourtant des mouvements lisses sur les figures 3.13 et 3.14), les seuils utilisés sont plus élevés et, de fait, le système devient moins sensible, et capture moins de lois sensorimotrices. Ainsi le système n'acquiert que peu de lois sensorimotrices concernant les covariations entre la proprioception, et les positions visuelles. Il faut toutefois noter que les deux SLEs de la main apprennent des contextes, spécifiques à des zones de la proprioception de θ_1 et de θ_2 et encodés par le SAW, dans lesquelles la main robotique ne peut plus bouger de par les butées articulaires.

Notons que même si tous les SLEs calculent leurs prédictions à chaque itération (ce qui a un coût computationnel élevé), ils apprennent indépendamment les uns des autres, et dans notre implémentation ils tournent en parallèle, et utilisent des communications asynchrones.

Nous décrivons ici la séquence des événements de l'expérience. La seule entrée externe de DSM (outre les senseurs et les moteurs), est un but, externe à DSM, relatif à la récompense (voir la figure 3.9).

Phase 1 : Atteinte de la cible. Nous donnons à DSM le but $Reward = 1$. En tant que but externe, il est associé à la priorité 1 ($P = 1$) dans le module de sélection de priorité (**PS**). La distance de la valeur courante, à la valeur de récompense 1, est calculée par le module de calcul de distance (**DC**), et est ensuite envoyée au SLS_{Reward} . Ce SLS envoie une commande motrice $\delta s_{H \rightarrow Target}$ au **PS**, associée avec une priorité $P = 2$, commande qui consiste de fait à approcher la main de la cible. Cela est réalisé à chaque itération de DSM. **DC** calcule les distances $\delta s_{H \rightarrow Target}$ sur les axes X et Y et les envoie à SLS_{H_x} et SLS_{H_y} , en les associant à une priorité $P = 2$. Les deux SLS envoient ensuite des commandes motrices sur θ_1 et θ_2 associées à $P = 3$. Ici, aucune autre commande motrice n'est envoyée, aussi le module de pondération par l'efficacité est inutile. Un comportement d'atteinte est alors observé. Puisque le SLE a appris qu'il y a un point où $\delta s_{H_y}^{Pred} < Th_2$, lorsqu'une cible est hors de portée, parce qu'elle est au-dessus de la main dans l'axe Y, le SLS vérifie donc lorsqu'il est dans ce cas si un S^{Sim} existe dans lequel $\delta s_{H_y}^{Sim} > Th_2$ (voir 3.2.3). Ce ne sera pas le cas si le SLE_{H_y} n'a pas encore appris la règle avec l'outil 1, et en ce cas le bras ne bouge plus et le robot reste en *phase 1*. Mais si la règle où l'outil 1 permet une variation sur l'axe Y a été apprise (ce qui peut se

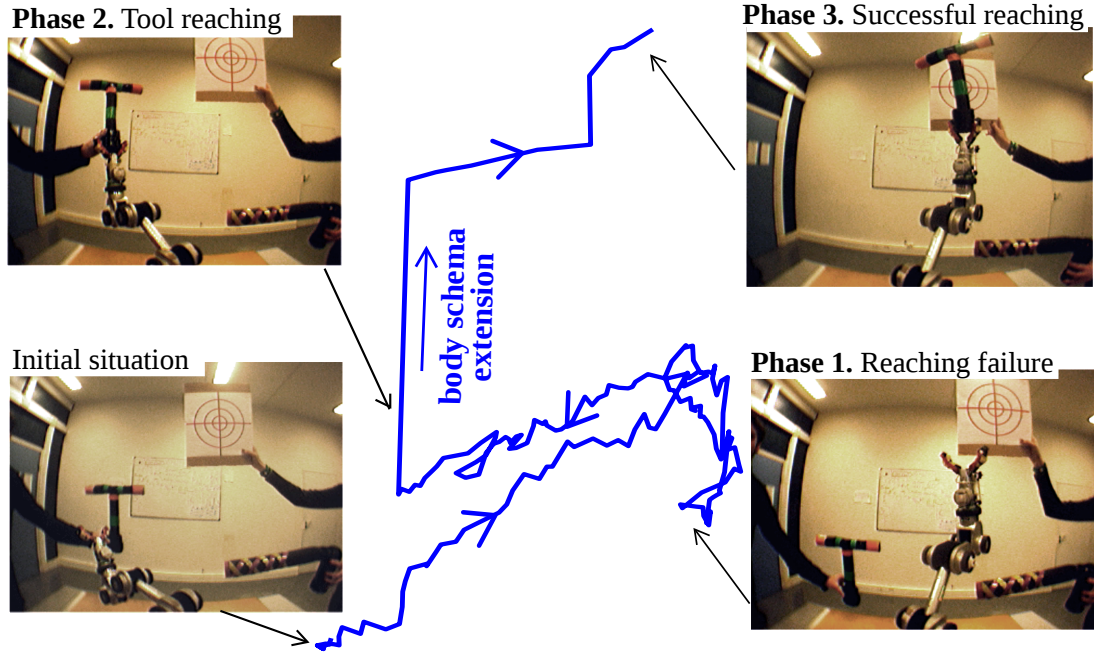


FIGURE 3.13 – Trajectoire de la main du bras Katana (telle que perçue par l’algorithme de reconnaissance de forme) avec la cible hors de portée selon l’axe Y. Le robot effectue un détour et saisit l’outil adapté, en l’occurrence l’outil 1.

produire à tout moment, indépendamment d’un apprentissage ou comportement prédéterminé), alors $\delta s_{H_y}^{S^{Sim}}$ peut avoir une valeur au-dessus du seuil. Ce sera typiquement le cas lorsque S^{Sim} correspond à la situation dans laquelle la main est suffisamment proche de l’outil 1. Le même mécanisme s’applique avec SLS_{H_x} et l’outil 2.

Phase 2 : Atteinte de l’outil. Nous décrivons ici le mécanisme de l’outil 1, mais il en est de même pour l’outil 2. Dans cette phase, la cible n’est pas accessible car elle est au-dessus de la main dans l’axe des Y, mais $\delta s_{H_y}^{S^{Sim}} > Th_2$ dans le SLS_{H_y} . Le SLS envoie un sous-but sur $s_{H \rightarrow Tool1}$ associé à $P = 3$. Cette fois, **PS** reçoit 2 valeurs souhaitées pour H_x et H_y , puisqu’elle a continué à recevoir $\delta s_{H \rightarrow Target}$, avec $P = 2$, de SLS_{Reward} . La valeur souhaitée avec la plus haute priorité est alors sélectionnée : **DC** calcule la distance de la main à l’outil 1 et les envoie à SLS_{H_x} et SLS_{H_y} , associé à $P = 3$. Comme en *phase 1*, le SLS envoie des commandes motrices et la main essaye d’atteindre l’outil 1. Notons que par contre, contrairement avec la *phase 1*, **PS** ne reçoit pas $s_{H \rightarrow Tool1}$ à chaque itération, mais seulement sur une itération, puisqu’après que SLS_{H_y} envoie seulement des commandes motrices. C’est pourquoi une **mémoire de travail** est nécessaire dans **PS**, afin de garder ce sous-but en mémoire jusqu’à sa réalisation.

Phase 3 : Atteinte de la cible avec l’outil dans la main. La main (pince du robot) saisit l’outil, et en suivant le principe de l’extension du schéma corporel, la main est perçue comme étendue à la position de l’outil sur les axes X et Y. Le sous-but est effacé de la **mémoire de travail** dans **PS**. Le système se comporte alors de la même manière que dans la *phase 1*.

Sur les figures 3.13 et 3.14, nous montrons les résultats d’expériences où les cibles sont inaccessibles sur les axes Y et X. Dans les deux cas, la tâche est la même, et consiste à augmenter du signal de récompense du robot, ce qui se traduit par une tâche d’atteinte. Chaque fois, les deux

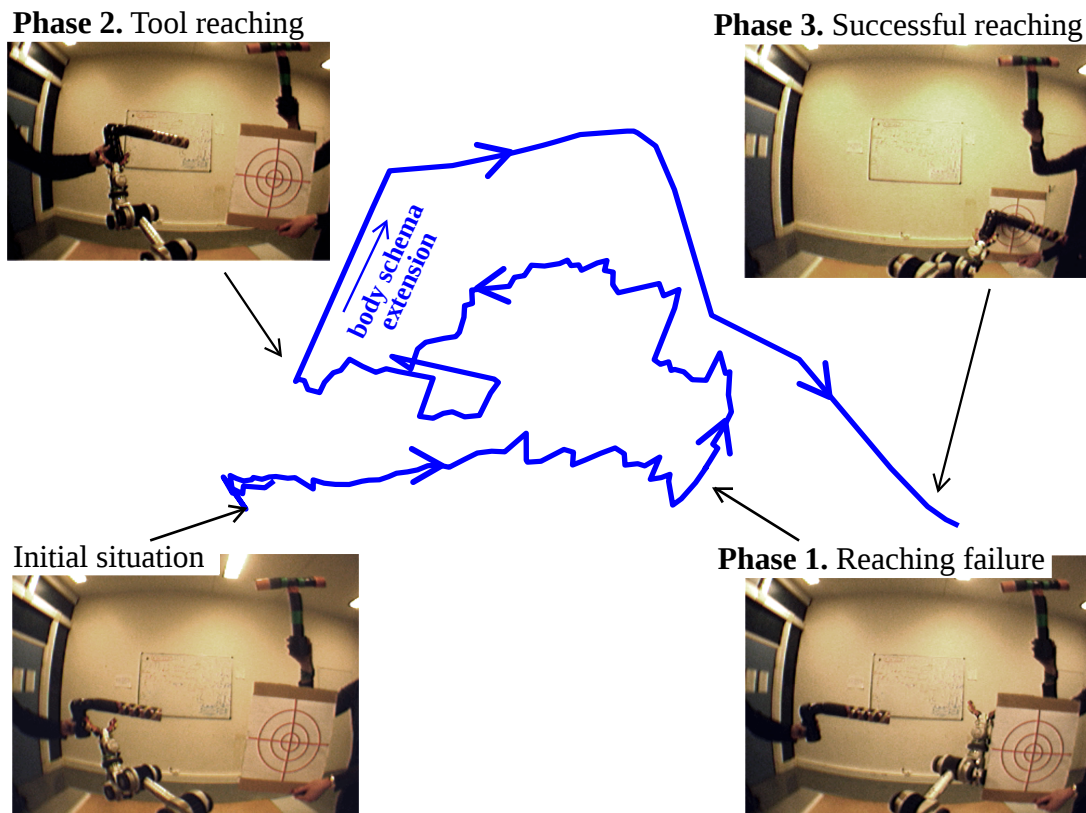


FIGURE 3.14 – Trajectoire de la main du bras Katana (telle que perçue par l’algorithme de reconnaissance de forme) avec la cible hors de portée selon l’axe X. Le robot effectue un détour et saisit l’outil adapté, en l’occurrence l’outil 2.

outils sont présentés au robot pendant toute l’expérience. Les résultats montrent la trajectoire du bras, telle que perçue par l’algorithme de reconnaissance de formes. Nous pouvons observer les 3 phases décrites ci-dessus : le robot essaie d’atteindre la cible, atteint une zone de sa proprioception dans laquelle il ne peut pas aller plus loin, le long de l’axe X ou Y, puis fait un détour pour sélectionner l’outil adapté à son problème.

Sur la figure 3.15, nous montrons des résultats plus détaillés d’une expérience réalisée seulement avec l’outil 1. En a), les positions de la main robotique (en bleu), de l’outil (en vert pointillé) et de la cible (en rouge avec des tirets) sont indiquées pour les axes X et Y, pendant le temps de l’expérience. En b), les commandes motrices données par θ_1 et θ_2 sont indiquées respectivement en bleu pointillé et en vert. En c), nous montrons les valeurs de priorité reçues par le SLS_{Reward} en rouge, par le SLS_{H_x} en pointillé bleu et par le SLS_{H_y} en vert pointillé. Ces deux SLEs ayant les mêmes priorités, les deux dernières courbes sont superposées. Au cours de la *phase 1*, le SLS_{Reward} envoie des valeurs désirées pour la main, afin que celle-ci atteigne la cible, avec une priorité de $P = 1$. Ces valeurs sont envoyées à SLS_{H_x} et SLS_{H_y} qui, en retour, envoient les commandes motrices désirées θ_1 et θ_2 avec pour priorité $P = 1 + 1 = 2$. Dans la figure 3.15 a), on peut ainsi observer que la main robotique (en bleu) essaie d’atteindre la cible (en rouge).

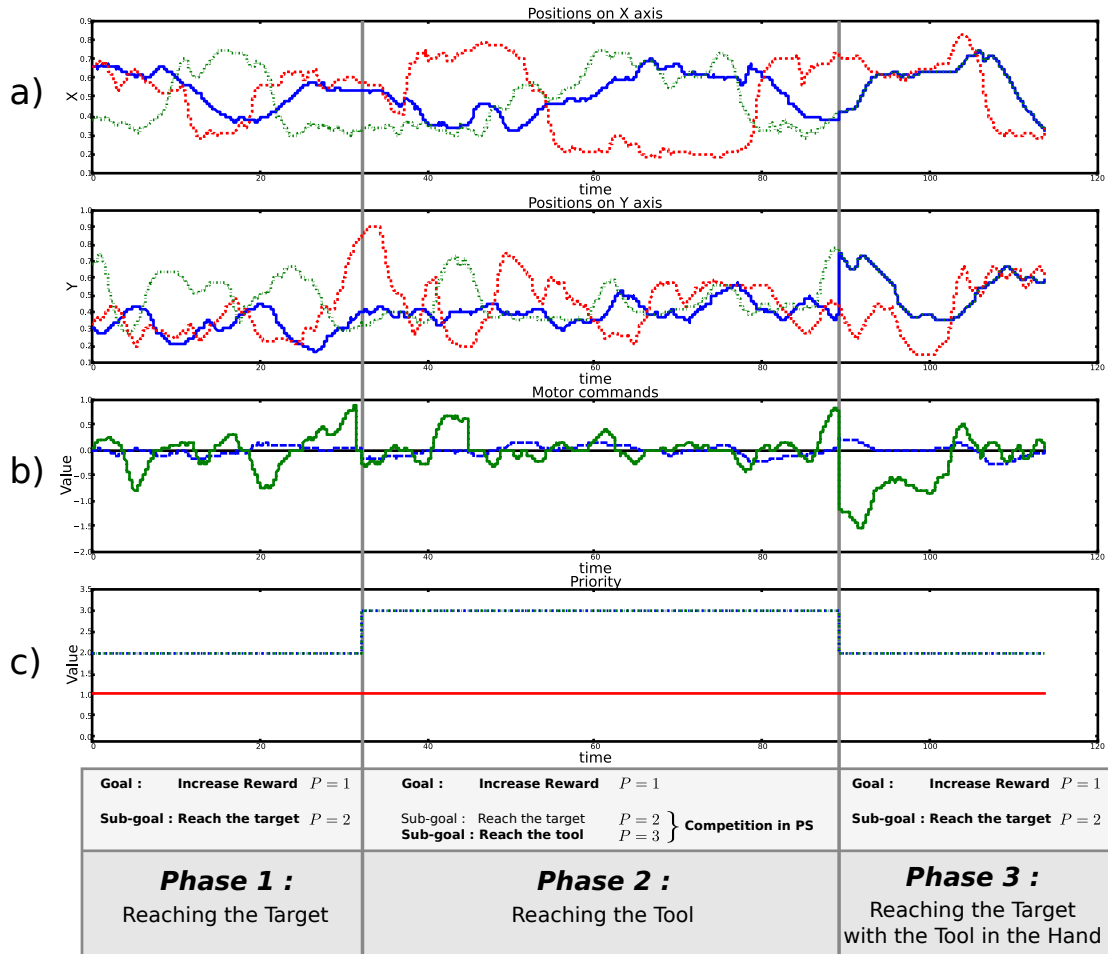


FIGURE 3.15 – Résultats de l'expérience avec l'outil 1. En a), les trajectoires telles que perçues par l'algorithme de reconnaissance de formes sont représentées sur les axes X et Y de l'entrée visuelle, en fonction du temps (en secondes). Celle de la main robotique est représentée en bleu plein, celle de l'outil en vert pointillé, et celle de la cible en rouge avec des tirets. En b), les commandes motrices concernant θ_1 et θ_2 sont représentées en tirets bleus et en vert, respectivement. En c), nous voyons les valeurs des priorités telles que reçues par le SLS_{Reward} en rouge, le SLS_{H_x} en bleu pointillé, et le SLS_{H_y} en vert pointillé.

Pour la deuxième phase, la cible est hors de portée sur l'axe Y et le bras est dans la zone de proprioception (apprise par la SAW du SLE) où il ne peut plus monter car il est en butée articulaire : $\delta s_{H_y}^{M^{Sim}} < Th_2$. En raison de l'équation 3.20 donnant la commande motrice en fonction de la valeur désirée et de la valeur simulée, laquelle est ici nulle, les commandes motrices sur θ_2 sont à 0 à ce moment-là (voir figure 3.15 b) en vert). Le SLS_{H_y} envoie alors un sous-but sur $s_{H \rightarrow Tool}$ associé à $P = 2 + 1 = 3$, puis SLS_{H_x} et SLS_{H_y} envoient des commandes motrices θ_1 et θ_2 pour atteindre l'outil avec une priorité de $P = 3$. Dans la figure 3.15 a), on peut ainsi observer que la main robotique (en bleu) essaie d'atteindre l'outil (en vert).

Une fois l'outil saisi, le sous-but est effacé de la mémoire de travail et, comme en phase 1, la valeur désirée provenant de SLS_{Reward} est de nouveau sélectionnée : le robot essaie d'atteindre

la cible mais cette fois avec l'outil en main. Dans la figure 3.15 a), on peut observer que la position de la main robotique (en bleu) et celle de l'outil (en vert) sont réunies, et essaient d'atteindre la cible (en rouge).

Le retard entre les commandes motrices et la position de la main est provoqué par les délais de l'algorithme de reconnaissance de formes. La différence d'amplitude entre les deux commandes motrices s'explique par les lois sensorimotrices apprises par SLS_{H_x} et SLS_{H_y} : avec le bras Katana, l'efficacité est plus élevée sur l'axe X que sur l'axe Y, au sens où pour un mouvement aussi grand sur chacun des axes, il faudra une magnitude motrice moindre pour le mouvement sur l'axe X. Rappelons également qu'en dernier lieu, le PID du bras change la commande réelle envoyée à l'effecteur.

3.2.5 Discussion et conclusion

Comme précédemment mentionné, dans DSM le SLE est le seul mécanisme d'apprentissage, dédié à la prédiction de variations sensorielles. Le SLS, à travers des mécanismes de simulation et d'inhibition, utilise lui ces prédictions pour percevoir des affordances. Ces affordances sont finalement utilisées pour calculer les actions à réaliser pour arriver à un objectif donné. La simulation motrice conduit à des commandes motrices désirées, quand la simulation sensorielle conduit elle à des configurations sensorielles désirées, à savoir des sous-buts. Ces valeurs désirées, émises par le SLS, sont inhibées ou prises en compte, grâce à la mise en œuvre d'un mécanisme de priorité. L'inhibition temporaire des buts précédents permet au robot d'effectuer des tâches intermédiaires, et ainsi de faire des séquences. Une mémoire de travail est responsable du stockage de sous-buts, lesquels n'auront pas à être en permanence émis par les SLS pour ne pas être oubliés. De la sorte, chaque SLS se contente uniquement d'envoyer les commandes motrices, ou sous-buts, en rapport avec la variation désirée courante. Ce faisant, les couples de SLE et SLS ne sont pas informés de la séquence, et ainsi de la hiérarchisation des actions en cours, alors même que celles-ci proviennent de ce que les SLE ont précédemment appris. Même sans planification explicite, cette approche permet l'apparition d'une cascade de sous-buts, calculés à la volée. Ainsi une séquence variée de sous-buts et d'actions peut apparaître en fonction du contexte de l'expérience, comme nous l'avons vu avec notre expérience, avec deux outils différents permettant de surmonter des difficultés différentes. Enfin, soulignons que ces séquences peuvent être réalisées, comme c'est le cas dans nos expériences, même si elles n'ont jamais encore été apprises, observées ou expérimentées.

De par cette manière de fonctionner, les séquences dépendent entièrement de ce que chaque SLE a appris pendant le développement du robot et des situations qu'il a rencontrées. Dans notre approche, les séquences ne sont pas supposées s'ajouter aux comportements élémentaires existants, mais sont les conséquences directes de certaines lois sensorimotrices spécifiques apprises. Par exemple, dans notre expérience le détour pour saisir l'outil est dû à la manière dont le robot a appris la loi de l'outil.

Si donc les entrées, déterminant l'apprentissage qui est fait, ont été choisies par le modélisateur dans notre expérience, on peut vouloir attendre d'un robot qu'il ait la capacité d'apprendre en continu de nouvelles lois, et de modifier les lois précédemment apprises.

De la même manière, il serait souhaitable que de nouveaux moteurs puissent être ajoutés sur les SLEs, et que de nouveaux SLEs puissent être créés et ajoutés à la volée. Or ceci est déjà possible. Il est possible d'ajouter des couples de SLE et SLS, car chaque couple n'est utilisé

que pour l'ajout de nouvelles commandes motrices, ou configurations sensorielles souhaitées, à celles qui se présente déjà au mécanisme de sélection de priorité. De plus, l'ajout d'un nouveau moteur au sein d'un SLE est également déjà possible, dans la mesure où cela ne consiste qu'en l'ajout d'une nouvelle contribution parmi les autres, dans le LMS (voir figure 3.1), lequel ne se contente que d'additionner des contributions. Cela permet un apprentissage incrémental, où les compétences acquises antérieurement ne sont pas supprimées, à mesure que de nouvelles sont apprises.

Dans le SLE, l'entrée S devrait également être flexible afin de s'accommoder de nouveaux senseurs de haut niveau (comme, dans nos expériences, de senseurs indiquant la présence et les coordonnées de différents objets, ce qui aura pourtant à être préalablement appris). Ceci pose problème pour le SAW actuel, mais cette question est traitée dans le chapitre suivant.

Comparaison avec des approches connexes

L'utilisation d'outils soulève le problème de l'adaptation cinématique. Dans [Rolf et al., 2010], un seul réseau neuronal récurrent est utilisé pour apprendre la cinématique inverse des différents outils, sans réapprentissage ou oubli du schéma corporel. Cependant, la longueur de l'outil est un paramètre de ce réseau neuronal, qui doit être connu à l'avance. Considérant ce problème, dans [Jamone et al., 2012] Jamone et collègues montrent comment la cinématique pourrait être adaptée à l'utilisation de l'outil quand aucune information n'est donnée sur l'outil saisi. Ils ont utilisé pour cela le modèle IMLE qui permet l'apprentissage incrémental de différentes solutions lors d'entrées similaires, et permet ainsi de représenter les relations d'un vecteur d'entrée à un vecteur de sortie dans des différents contextes inconnus spécifiques. Pendant l'apprentissage et la phase d'utilisation, l'algorithme traite automatiquement d'un ensemble approprié de modèles linéaires locaux, et cette connaissance est exploitée pour une tâche d'atteinte.

Cependant, Nabeshima et collègues [Nabeshima et al., 2006] défendent l'idée que le véritable problème pourrait plutôt être celui de la détection des altérations corporelles, basées sur les données sensorielles. Ils proposent ainsi un modèle pour adapter le schéma corporel du robot à un outil. Le robot (un bras à 2 degrés de libertés dans le plan, et une caméra) apprend les associations entre l'intégration spatiale et temporelle de l'information visuelle et tactile, lorsque la main du robot touche une cible. L'association est ensuite stockée dans une mémoire associative. Ainsi, lorsque le robot touche la cible avec cette fois un outil (un bâton), une nouvelle association entre les informations visuelles et tactiles est apprise dans la mémoire associative, et le contrôleur cinématique du bras est adapté.

Bien que les résultats préliminaires aient été donnés dans la section précédente (3.1.4), l'adaptation du schéma corporel à l'utilisation de l'outil n'est pas ici le sujet central auquel nous nous sommes intéressés en tant que tel. Cependant, ceci est considéré comme un sujet à approfondir dans de futurs travaux, notamment au travers l'utilisation d'un nouvel outil de catégorisation des données sensorielles (voir chapitre 4). Concernant le problème soulevé par Nabeshima et collègues, notons que le processus d'apprentissage du SLE est directement dédié à la détection des altérations des prédictions (et donc dans le cas des prédictions corporelles, comme le mouvement de la main, des altérations du corps) sur la base des données sensorielles. De plus, cette détection repose sur l'erreur accumulée pendant la durée d'un mouvement (jusqu'à ce que δs dépasse un seuil). Cependant, la principale différence avec leur approche repose sur le fait que nous nous concentrons moins sur la manière d'acquérir de telles connaissances que sur la manière dont celles-ci peuvent être utilisées pour changer le comportement du robot,

en l'occurrence dans une tâche d'atteinte, en lui permettant de par lui-même faire un détour pour se saisir d'un outil adapté à sa tâche.

Dans [Billing et al., 2015], des champs de neurones dynamiques (DNF, voir [Schöner, 2008] et la section 1.1.4) sont proposés pour combler le fossé entre les dynamiques sensorimotrices bas niveaux et des processus cognitifs plus hauts niveaux, tel que la planification. Les auteurs proposent des comportements élémentaires tels que “chercher un objet de la couleur X” ou “bouger le bras à la position Y” associés avec des DNF conditionnant l'initialisation et la finalisation (conditions de satisfaction) de ces comportements. Chaque comportement est aussi associé avec un champ de motivation (permettant de lancer le comportement) et un champ de précondition (pour savoir si le système est à même de le faire). Une fois que le champ de motivation est activé, le DNF gérant l'initialisation est aussi activé mais peut être inhibé si le champ de précondition ne satisfait pas les conditions nécessaires. Dans ce cas, ce champ peut activer le champ de motivation d'un autre comportement élémentaire qui permettra de remplir cette condition. Une fois celle-ci effectivement satisfaite, le champ de précondition arrête l'inhibition du comportement.

Ce mécanisme est proche de celui décrit dans DSM. En effet, de la même manière notre couche de catégorisation dans le SLE est utilisée (par simulations) pour envoyer la configuration sensorielle désirée (ce qui est l'équivalent d'une motivation dans le système que nous venons de décrire), et ce lorsque les commandes motrices ne sont pas suffisamment efficaces pour la tâche en cours (ce qui est le cas lorsqu'il n'y a pas d'affordance), ce qui est l'équivalent du champ de précondition. Une commande motrice est alors exécutée (équivalente au champ d'initiation) lorsque la configuration sensorielle désirée est atteinte (alors équivalente à la condition de satisfaction). La différence principale réside dans le fait que les sous-tâches ne sont pas initiées ou terminées de la même manière : dans DSM, l'équivalent à la condition de satisfaction n'est pas donné par la sous-tâche, mais par le SLS qui aura en premier lieu envoyé le sous-but désiré. Le problème de l'initiation et de la fin d'une sous-tâche n'est alors pas lié à la réalisation d'un comportement spécifique, mais aux seules lois sensorimotrices apprises (et plus précisément encore, à la couche de catégorisation du SLE). La requête n'est pas sur un comportement, mais directement sur une configuration sensorielle de bas niveau. Notre approche, qui relie directement les compétences sensorimotrices de bas niveau à des capacités de séquence de haut niveau, permet de fait à ces séquences d'être très plastiques. De fait, n'importe quel SLS peut envoyer une configuration sensorielle désirée, et créer ainsi une séquence spécifique à la volée, et ce sans planification. Grâce à la mémoire de travail externe aux SLSs, ceux-ci sont capables de modifier indirectement leur propre but courant, comme cela a été illustré dans les expériences d'utilisation d'outils ci-dessus.

Dans [Baranes and Oudeyer, 2013], les auteurs décrivent un modèle d'apprentissage d'associations entre des compétences motrices paramétrées, et les tâches paramétrées correspondantes, basé sur une mesure des progrès de la compétence. Ils proposent d'optimiser la capacité du robot à résoudre des tâches à partir d'une curiosité artificielle, et le robot explore l'espace de travail au lieu d'effectuer un babillage moteur. Cependant, cette approche ne se concentre pas sur la capacité du robot à résoudre de nouvelles tâches, ou de les résoudre d'une nouvelle manière.

Dans [Hart and Grunewald, 2011], un ensemble de contrôleurs dynamiques en boucle fermée est relié à une machine à état fini dont les états activent un ou plusieurs contrôleurs. Chaque contrôleur est piloté par une fonction sensorimotrice primitive. Les transitions d'état sont déclenchées par la convergence de chaque contrôleur vers un attracteur stable. Ces transitions sont apprises par le biais d'un Q-learning en utilisant en récompense une motivation intrinsèque,

laquelle détecte la convergence (appelée quiescence) vers lesdits attracteurs. Il en résulte un système capable de découvrir des affordances, définies par les zones sensorielles spécifiques où chaque comportement est stable, et ce système peut générer des séquences complexes en suivant les règles de transition apprises, guidées par la convergence des contrôleurs. Notons que cette approche ne prédit pas directement les conséquences sensorielles des actions, mais exploite plutôt des critères de stabilité comportementale pour amener à l'activation de contrôleurs permettant de réaliser une tâche particulière.

Dans [Law et al., 2014], les auteurs étudient des systèmes d'explorations "open-ended", et des comportements de recherche de buts, et privilégient les comportements de babillage moteur (ou encore de jeu) pour stimuler l'apprentissage, plutôt que les méthodes axées sur des comportements guidés par un but. Les auteurs sont intéressés par l'évolution du comportement lié aux transitions entre différents stades du développement : ils montrent comment l'évolution de contraintes sur le robot, inspirés de la psychologie développementale, aident à réduire la complexité de l'apprentissage à chaque étape du développement, et comment l'apprentissage acquis au cours de précédentes étapes amorce et simplifie les étapes ultérieures. L'apprentissage est basé sur des associations de paires entre senseurs et de moteurs, il s'agit donc d'une stratégie que nous avons appelée absolue.

Dans [Perotto, 2013], Perotto a proposé un modèle (CALM) où les régularités sont apprises dans des schémas composés de trois vecteurs : contexte + action \rightarrow attente. Chaque vecteur est composé de valeurs discrètes, en opposition avec notre modèle où le contexte, le moteur et la variation attendue du senseur sont tous continus. Cependant, ces états peuvent contenir des valeurs non définies, qui offrent ainsi de bonnes propriétés de généralisation. L'apprentissage est incrémental, et les schémas peuvent devenir de plus en plus précis, ou encore être corrigés après de nouvelles expériences, voire fusionnés avec d'anciens schémas si nécessaire. De plus, les valeurs traitées ne sont pas nécessairement observables : elles sont alors dites "synthétiques", et permettent de traiter des environnements partiellement inobservables, et d'abstraire des propriétés utiles (elles ont notamment été conçues pour faire le pont avec des propriétés symboliques de haut niveau), ou encore d'identifier les différentes étapes dans une séquence d'action. Dans DSM, des équivalents en termes de senseurs ou moteurs synthétiques, ou cachés, pourraient être utilisés pour traiter les environnements partiellement inobservables bien que déterministes. Enfin, notons que le modèle CALM prédit les états finaux mais non les changements opérés, ce qui pourrait offrir de meilleures propriétés de généralisation.

Dans [Ugur et al., 2011], Ugur et collègues utilisent l'apprentissage des affordances pour la planification. Dans leurs travaux, le robot apprend au cours d'une phase de babillage des associations entre des caractéristiques initiales d'un objet perçu, et des effets sur ces caractéristiques après un comportement donné. L'effet est la différence entre les caractéristiques initiales et finales de l'objet. Puisque les états finaux peuvent alors être calculés en ajoutant l'effet à l'état en cours, il est possible de créer un arbre de recherche, avec des nœuds composés d'états perceptuels, et des arêtes composés de paires comportement-objet. Pour un but donné, le robot crée un tel arbre et l'étend graduellement, en partant de l'état qui est à une distance minimale du but, afin de sélectionner une séquence dans l'arbre permettant d'atteindre l'état satisfaisant l'objectif. Cette approche peut être considérée comme proche de notre travail, en ce que le robot apprend des associations entre des contextes sensoriels continus, et les effets de l'action, et trouve une séquence pour minimiser la distance du but. Cependant, il existe des différences importantes qui distinguent notre travail. Tout d'abord, dans DSM chaque entrée (senseurs et moteurs) et sortie

(variation du senseur) du SLE est continue. Les commandes motrices ne sont ni discrètes, ni définies à l’avance en tant que comportements pré-codés, comme c’est le cas dans leur approche. Dans DSM, il s’agit de magnitudes motrices de bas niveau, ou encore de variations des senseurs que le modèle est à même de prédire, grâce à la propriété hiérarchique. Deuxièmement, la catégorisation effectuée dans DSM n’est pas déterminée par la spatialité du senseur ou sa topologie, mais elle est déclenchée par une erreur de prédiction dans les lois sensorimotrices utilisées, lors du babillage (voir la section 3.1.1). La catégorisation dépend ainsi de l’action effectuée, et uniquement des données sensorielles. Les catégories sont uniquement utilisées pour contextualiser les lois sensorimotrices. Troisièmement, dans DSM l’espace des effets (les dérivées de senseurs) et les espaces moteurs (magnitudes motrices) ne sont pas catégorisés, mais linéairement associés. Et ces associations linéaires, qui constitue notre implémentation du principe de codage commun, sont utiles pour considérer l’espace (simulé) des contextes comme un espace homogène à celui des moteurs, et donc ainsi dans lequel les actions motrices sont immédiatement possibles. Autrement dit, plus le robot apprend les effets de ses actions, plus il apprend à percevoir ses actions comme étant des variations de contextes, ce qui lui permet, quand nécessaire, d’obtenir l’ensemble des caractéristiques contextuelles requis pour l’accomplissement d’une tâche.

Enfin, dans DSM, il existe des séquences sans module dédié à la planification : le mécanisme décrit permet d’obtenir des sous-buts sans arbres de recherche, lesquels sont basés sur des états discrets. La planification provenant du DSM considère directement des commandes motrices de bas niveau en sortie, et non une liste de scripts comportementaux. À la place d’un chaînage basé sur des états discrets, notre modèle propose des séquences créées à la volée, où chaque SLE concerné est indirectement capable d’en utiliser d’autres pour atteindre son sous-but, qui sera alors exécuté avec une priorité plus élevée. À la différence de [Ugur et al., 2011], et des méthodes similaires qui dépendent de chaînages en avant, la séquentialité émergente de DSM suit une approche de chaînage inverse, ce qui permet d’éviter les problèmes d’exploration combinatoire qui y sont liées. Soulignons ici que même si notre modèle ne permet que de “planifier” une seule étape en avant, par la simulation de l’étape suivante, cela donne déjà lieu à la création d’une séquence impliquant des sous-objectifs.

Dans [Chaput, 2004], Chaput propose un modèle constructiviste appelé CLA basé sur une propriété hiérarchique. Comme dans notre modèle, cette propriété hiérarchique vient du fait que dans CLA les schémas sensorimoteurs ont des entrées et des sorties avec le même type de représentation. La première couche est composée de senseurs bruts, et les couches supérieures suivantes ont la sortie d’une couche inférieure en entrée. Mais Chaput a également proposé un mécanisme intéressant exploitant ces propriétés hiérarchiques : la couche en entrée pourrait être sujette à un retard temporel, ce qui peut conduire à des structures récurrentes. Même si lorsqu’une nouvelle couche est créée, la couche de niveau directement inférieur est plus susceptible d’être sélectionné en entrée, il suggère un mécanisme de recours permettant de se rabattre, dans certains cas, sur une entrée choisie parmi les couches les plus basses, notamment lorsque le robot reçoit un stimulus pour lequel il n’a jamais encore été entraîné. Les résultats de Chaput sont considérés comme de possibles pistes permettant d’améliorer DSM, notamment dans la manière de considérer les entrées motrices, et même sensorielles, à prendre en compte en entrées de nouveaux SLEs.

Les “Complexes Action Objet” (OAC) sont conçus pour combler l’écart entre les représentations sensorimotrices de bas niveau, requises pour la perception et le contrôle du robot, et les représentations de haut niveau nécessaire pour le raisonnement et la planification abstraits [Kri-

ger et al., 2011]. Sur la base d'états discrets, le modèle apprend les états finaux en cas d'exécution d'un programme moteur (par exemple, des actions réflexes codées en dur, provoquées par des entrées visuelles). Mais ces états contiennent toutes les informations, celles pertinentes comme celles non pertinentes, pour la prédiction. L'objectif du module d'apprentissage est alors de soustraire les informations non pertinentes pour avoir des états comportant uniquement les informations pertinentes nécessaires pour prédire efficacement les effets de l'action. Ce problème existe dans le contexte de DSM et est similaire à celui décrit dans [Krüger et al., 2011], dans la mesure où tous les senseurs pourraient potentiellement être pertinents pour prédire une variation de senseur. Ainsi DSM aurait idéalement à abstraire les entrées nécessaires, non seulement pour être plus précis, mais surtout pour envoyer des sous-buts ne contenant que les configurations sensorielles pertinentes. Dans nos expériences nous avons présélectionné les entrées pertinentes, mais ce travail devrait être effectué par DSM lui-même et non par le modélisateur. Cet aspect sera développé dans le chapitre suivant.

Propriétés provenant du principe idéomoteur

Dans la lignée du principe idéomoteur, DSM permet de percevoir les buts comme un ensemble de distances sensorielles traduisible en actes moteurs, et donc d'agir grâce à l'anticipation des conséquences de l'action. Pour obtenir cela, le SLE est une proposition pour percevoir des variations sensorielles dS en tant qu'équivalentes à des magnitudes motrices M (en supposant que des relations linéaires locales existent entre elles).

Comme mentionné précédemment, la magnitude motrice d'entrée d'un SLE peut être une variation sensorielle donnée dS . Dans la mesure où cette dernière peut être prédite par un autre SLE, celui-ci peut être considéré d'un "niveau inférieur" par rapport à celui qui utilise sa sortie en entrée motrice. Dans notre expérience, nous avons utilisé $\delta s_{H \rightarrow Target}$ comme entrée motrice pour SLE_{Reward} . Cette caractéristique propre à DSM met en évidence deux propriétés intéressantes :

Premièrement, il permet à l'apprentissage du robot d'être hiérarchique, au sens donné par Chaput, c'est-à-dire que les premiers SLEs auront des senseurs "bruts" en entrée, quand les SLEs de niveaux supérieurs pourront privilégier, comme entrées motrices, des variations de senseurs si par exemple celles-ci sont suffisamment correctement prédites par des SLEs de niveaux inférieurs. Dans les expériences présentées par exemple, cette fonctionnalité permet d'apprendre des lois, éventuellement complexes, qui existent entre le mouvement du bras et l'environnement (interactions physiques ou sociales). De telles lois, qui peuvent être considérées comme de haut niveau, seront pour autant détachées de la complexité du seul apprentissage, bas niveau, des lois de déplacement du bras sur lesquelles elles reposent pourtant. Car ces lois seront apprises par d'autres SLEs (SLE_{H_x} et SLE_{H_y} dans notre expérience), et seul le résultat de ces apprentissages (une prédiction suffisamment correcte pour considérer la variation de senseur prédite comme potentielle magnitude motrice) pourra être utilisé par tout autre SLE. Grâce à ce mécanisme, il est possible d'utiliser le SLE pour des prédictions de très bas niveau, mais aussi, progressivement, pour des prédictions de plus haut niveau grâce à une construction hiérarchique.

Deuxièmement, puisque l'entrée motrice peut être une variation sensorielle, un apprentissage par observation est rendu possible, dans la mesure où un SLE peut utiliser l'observation d'une variation sensorielle à la place d'une activité motrice réelle. Comme envisagé précédemment, étant donné que de nouveaux SLEs peuvent être ajoutés en ligne sans modifier le système, il est à envisager qu'une variation sensorielle puisse être considérée comme entrée motrice d'un

SLE, dès qu'il existe un SLE prédisant cette variation sensorielle suffisamment correctement (les travaux de Chaput sont également à considérer, dans la perspective d'un biais permettant de présélectionner un sous ensemble d'entrées motrices). Dans un tel scénario, on pourra dans de futures expériences proposer qu'un apprentissage par observation soit en droit de se produire dès lors que le robot a découvert les lois sensorimotrices lui permettant de reproduire par lui-même ce qui observée.

Limitations et conclusion

La capacité à faire des séquences de notre modèle pourrait être questionnée du point de vue de son développement, ou des effets que pourrait engendrer un mauvais fonctionnement de certaines de ses composantes.

Si l'ensemble des mécanismes permettant d'effectuer des sous-buts (ou bien la mémoire à court terme) met du temps à se développer, les compétences atomiques (propre à chaque SLE) pourront être acquises sans qu'elles puissent être chaînées correctement. Il faudrait donc du temps pour passer de l'accumulation de ces compétences individuelles, à la capacité à les séquencer en vue d'un but, comme c'est le cas chez le bébé (voir la section 1.1.1). De manière proche, si par exemple le mécanisme permettant de sélectionner les priorités venait à défaillir, s'ensuivrait un ordre d'enchaînement des actions non pertinent pour satisfaire un but donné, quand bien même chacune de ces actions puisse être nécessaire. Or ce comportement serait alors proche de ce que l'on peut retrouver dans l'apraxie idéatoire (voir [Osiurak et al., 2010], et la section 1.1.1).

Comme indiqué précédemment, dans notre approche, une séquence dépend entièrement des lois sensorimotrices que le robot a apprises lors de son développement, par exemple au cours d'une phase de babillage (et pas nécessairement en exécutant des séquences, ou encore en essayant d'exécuter une tâche spécifique). Ce sont les entrées sensorielles du SLE qui sont utilisées lors de simulations du SLS. De sorte que toute configuration sensorielle désirée, envoyée par un SLS_i , contient toute entrée sensorielle, pertinente ou non pertinente, du SLE_i (voir la section 3.2.2). De fait, lorsque le système est suffisamment mature pour faire des séquences, celles-ci ne dépendront que de l'apprentissage qui aura été fait par la couche de catégorisation du SLE. Par exemple, si la couche de catégorisation ne fait pas de distinction entre avoir un outil ou non dans la main, alors une simulation avec n'importe quel senseur d'entrée simulé ne pourra jamais aboutir.

Le problème de distinguer les contextes pertinents, pour une prédiction, est renforcé par le fait que, du point de vue du développement, de nouveaux senseurs peuvent être ajoutés en ligne, et que potentiellement tous les senseurs peuvent se révéler pertinents (en particulier dans le cas de l'utilisation d'un outil). Actuellement, si le modèle reçoit des senseurs d'entrée non pertinents, ils participeront à part égale aux mécanismes internes du SLS, et donc aux configurations désirées en sortie, dont un sous-ensemble pourra de fait être aléatoire. En outre, étant donné que de nombreuses couches de catégorisations sont nécessaires (éventuellement un par couple de moteur et de capteur), les entrées non pertinentes peuvent augmenter rapidement le coût de calcul. Ces problèmes seront traités dans le chapitre suivant.

Dans notre approche, la manière de catégoriser les entrées sensorielles détermine ce qui sera prédicteur de variations sensorielles, et donc en dernier lieu spécifie des affordances particulières, au sens où elle détermine la perception qu'a le robot de sa capacité à agir sur le monde. Les affordances sont usuellement rattachées aux propriétés visuelles des objets. Ce problème

n'a pas été abordé ici en tant que tel, toutefois, une piste de recherche serait, avec DSM, de catégoriser les "bonnes" primitives visuelles en entrée de la couche de catégorisation du SLE : en exploitant le lien direct entre ce qui est catégorisé dans le SLE, et les affordances menant aux actions, un SLS pourrait envoyer des valeurs désirées sur ces primitives visuelles. Alors, tout objet ayant ces propriétés pourrait être "désiré". Par exemple, dans notre expérience, si nous utilisons en entrée de SLE_{Hy} , des primitives visuelles capables de distinguer un objet allongé, au lieu de la reconnaissance de l'outil (qui suppose de plus l'apprentissage de cet objet particulier), alors on peut attendre que suite à un apprentissage ayant permis d'isoler les entrées sensorielles pertinentes, lors d'une tâche d'atteinte du type que nous avons effectué le robot désirera atteindre un objet allongé dans son environnement et non précisément l'objet appris, et pourrait alors ainsi s'adapter à divers objets.

De plus, la théorie des affordances suggère que les propriétés visuelles des objets sont associées à la manière spécifique dont ceux-ci doivent être saisis. Par exemple, si la poignée d'une tasse est sur la droite, la main robotique doit l'approcher par la droite. Une telle planification pose problème avec des méthodes de descente en gradient comme la nôtre, car dans notre modèle le robot agit uniquement dans le but de réduire la distance de la main robotique à l'objet ciblé, sans autre considération. Un problème similaire peut se poser en présence d'obstacles qui doivent être contournés. Cependant, la capacité de DSM à créer des séquences à la volée, basées sur des affordances immédiates et différées, peut être combinée avec des boucles fermées de simulations internes avec les SLEs, afin pour résoudre de tels problèmes. Cette idée est développée dans [Srinivasa et al., 2012], dont le modèle utilisé est proche du SLE, et où ce qui est appelé répétition mentale est utilisé pour s'attaquer au problème de l'évitement d'obstacles. De telles boucles de simulations internes sont également considérées comme de futurs travaux.

- “Le mot [...] n’existe que par sa relation et son opposition avec les mots qui lui sont associés.”
- “Dans l’intérieur d’une même langue, tous les mots qui expriment des idées voisines se limitent réciproquement : des synonymes comme redouter, craindre, avoir peur n’ont de valeur propre que par leur opposition ; si redouter n’existait pas, tout son contenu irait à ses concurrents.”
- “Quand on dit que [les valeurs des mots] correspondent à des concepts, on sous-entend que ceux-ci sont purement différentiels, définis non pas positivement par leur contenu, mais négativement par leurs rapports avec les autres termes du système. Leur plus exacte caractéristique est d’être ce que les autres ne sont pas.”

De Saussure, *Cours de linguistique générale*, 1916

CHAPITRE 4

Apprentissage de lois causales incrémentales

Dans la partie précédente nous avons, avec le modèle sensorimoteur dynamique (ou DSM, *Dynamic Sensorimotor Model*), encodé des lois sensorimotrices linéaires qui apprenaient comment des activations motrices entraînent des variations de valeurs de senseurs. Ces lois étant variables, nous avons contextualisé localement les non-linéarités de ces lois à mesure que l’espace sensorimoteur était parcouru. Un contexte était donc un ensemble de valeurs de senseurs déterminant une loi linéaire particulière. Nous avons par la suite utilisé un mécanisme de simulation pour obtenir un modèle inverse qui ne soit pas appris mais créé à la volée en fonction du contexte courant, et nous avons également utilisé la simulation pour générer des sous-buts basés sur ces contextes. La pertinence des sous-buts demandés dépendait donc de celle des contextes : afin de tester le principe général de DSM, nous avons donc, dans le chapitre précédent, contourné cette difficulté en choisissant nous-même les entrées pertinentes, autrement dit les entrées dont les variations entraînent un changement dans la loi observée (ce qui peut être rattaché à la notion de variable explicative en statistique). L’un des objectifs de ce chapitre est de proposer une solution au problème du choix des entrées pertinentes, dans la lignée des principes de DSM.

Rappelons l’a priori à la base de ce modèle, selon lequel toute variation, et en premier lieu celles du monde environnant, perçues par les senseurs, est due à une activation motrice du robot (voir sec. 3.1.1). Ce dernier aura donc pour tâche de déterminer ces lois de co-variations en associant linéairement les variations motrices aux variations sensorielles. Bien entendu il s’agit d’un a priori fort, puisque le monde extérieur est sujet à des variations qui ne sont pas le fait du robot. Si cela doit être pris en compte dans des évolutions futures de notre modèle, notre démarche consiste, dans cette thèse, à pousser cette idée pour voir en quoi elle peut toutefois constituer un principe d’apprentissage sensorimoteur pertinent.

Ce principe selon lequel toute variation est due à une activation motrice du robot, est dans ce chapitre étendu aux variations mêmes des lois sensorimotrices. On supposera donc que toute variation de loi sensorimotrice (c’est-à-dire toute variation de loi de co-variations linéaires entre senseurs et moteurs) est elle-même due à une activation motrice du robot. Ceci se justifie par le fait que le robot est en mesure d’explorer lui-même son espace sensorimoteur, et donc d’explorer, activement, chaque frontière de ses lois sensorimotrices. Selon le principe que nous avons énoncé, l’enjeu est de déterminer quel est le “moteur” de la variation de la loi. Il s’agit donc

du moteur de la variation de contexte, au sens de contexte que nous avons défini précédemment (un ensemble de valeurs de senseurs), puisque le contexte détermine justement la loi applicable. Or, rappelons que l'un des principes fondamentaux de DSM est de percevoir des variations de senseurs comme des activations motrices, et une variation de contexte est justement une variation de senseurs puisqu'un contexte est un ensemble de valeur de senseurs. En conséquence, une variation de contexte peut être perçue comme une activation motrice. (Notons qu'en ce sens, il s'agit du type de "moteur" à l'œuvre lorsqu'un sous-but est effectué, qui est ainsi à l'origine des séquences d'actions).

Nous proposerons, dans ce chapitre, une extension de DSM, dans laquelle seul l'encodeur de lois sensorimotrices (ou SLE, *Sensorimotor Law Encoder*) sera modifié, en remplaçant ce qui faisait contexte (le SAW suivi du WTA) par une nouvelle définition de contexte, et en associant celui-ci non plus à une loi sensorimotrice donnée mais à une variation (qui sera une correction) des lois précédemment apprises. Dans ce chapitre, ce qui fera contexte ne sera plus un ensemble de valeurs de senseurs localement associés à une loi sensorimotrice donnée, et ayant une loi d'activation locale sur des senseurs. Le contexte que nous définirons dans le présent chapitre rendra compte de la variation des senseurs suffisante pour donner lieu à un changement de loi donnée. Sa loi d'activation sera globale, et interprétable, dans le cadre que nous avons défini, comme étant une activation motrice (et en l'occurrence, nous le verrons, bornée entre 0 à 1). Ainsi, un contexte *encodera la frontière entre deux lois sensorimotrices, codera l'incrément nécessaire pour passer à la seconde loi depuis la première*, et enfin, en suivant le principe que nous nous sommes donné, *aura une activation qui pourra être interprétée comme le "moteur" du changement de loi*.

Enfin, ces changements n'empêcheront en rien l'usage des simulations permettant d'atteindre des valeurs de senseurs spécifiques, tel que présenté dans le chapitre 3. Au contraire, ce mécanisme pourra être utilisé pour améliorer l'apprentissage de ces nouveaux contextes. En attribuant la responsabilité de la variation d'une loi à la présence d'une frontière (qui fait contexte) dans l'espace des senseurs, on sera en mesure de biaiser notre exploration sensorimotrice afin de jouer, activement, sur les variations de ces senseurs pour déterminer ce qui précisément fait contexte.

En robotique développementale, le fait de biaiser son exploration pour améliorer son apprentissage pose la question de la motivation intrinsèque. Ainsi, de nombreux travaux visent à déterminer une manière d'explorer son espace sensorimoteur permettant de maximiser les progrès de l'apprentissage qui en est fait [Oudeyer et al., 2007; Ngo et al., 2013]. Cette question peut également être liée à la question du jeu chez l'enfant, qui est décrit comme une motivation intrinsèque pouvant être source de plaisir, et où des actions peuvent être répétées, avec des variations ou des ré-ordonnancements, et sans but apparent : les moyens permettant d'atteindre un but semblent plus important que le fait d'atteindre ce but [Smith, 2009; Pellegrini, 2009; Lee, 2011]. Selon Bruner, le jeu est lié au développement de la flexibilité de la pensée : il offre des occasions d'expérimenter les possibilités, et de mettre ensemble différents éléments, de différentes manières [Whitebread, 2010].

En gardant cette définition du jeu en tête, ajoutons que, selon Meltzoff et Gopnick, les nourrissons ne sont pas seulement des observateurs passifs des conséquences de leurs propres actions, mais peuvent intervenir, et, à la manière de scientifiques, postuler certaines causes cachées puis agir en conséquence en testant leurs prédictions. Ils peuvent ainsi tenter d'inférer les relations causales entre différents événements en essayant de démêler, parmi des corrélations observées

entre événements, celles dues au hasard de celles dues à de véritables relations causales [Gopnik et al., 2000, 2004; Meltzoff, 2007].

Dans ce chapitre la capacité de DSM à faire des séquences sera utilisée pour mettre en jeu les senseurs qui, en variant, sont supposés avoir causé le changement d'une loi sensorimotrice. Des séquences seront donc effectuées en activant le ou les "moteurs" (au sens de contexte donné plus haut, c'est-à-dire en faisant varier des senseurs) afin de tester ce qui est à l'origine de la variation d'une loi. L'idée est qu'en faisant l'hypothèse que la cause (du changement de loi) peut être saisie grâce aux expériences sensorimotrices passées du robot, et plus précisément entre le moment où ses prédictions sont correctes et celui où elles échouent, le robot se trouvera motivé à bouger afin d'atteindre différentes configurations sensorielles spécifiques. Ce comportement, que nous appelons jeu, lui permettra ainsi de tester ses hypothèses sur ce qui définit la frontière observée, et donc sur ce qui précisément fait contexte, dans la nouvelle définition que nous avons donnée de ce mot (à savoir une variation de senseurs entraînant un changement de loi).

Pour savoir précisément comment nous allons faire ceci, nous allons dans un premier temps (Sec. 4.1) formaliser quelques propriétés des entrées de notre système, et présenter plus en détails le principe de notre proposition. Dans une seconde partie (Sec. 4.2), nous présenterons notre extension du modèle DSM et son intégration au sein de celui-ci. Enfin, dans une dernière partie (Sec. 4.3), nous présenterons une série d'expériences visant à expliciter, et à démontrer la viabilité de cette proposition.

4.1 Analyse du problème

4.1.1 Propriétés du système étudié

Lois imprévisibles : les entrées sensorielles sont toutes potentiellement pertinentes

Comme nous l'avons souligné précédemment, le modélisateur ne devrait idéalement pas avoir à choisir les entrées pertinentes pour la prédiction d'un sensor particulier (et donc en entrée du SLE prédisant cette variation). Car cette liste, en plus d'être potentiellement très exhaustive, est destinée à être enrichie au cours du développement et des circonstances auxquelles le robot sera confronté.

Prenons dans une expérience de pensée, l'exemple des mouvements d'un point caractéristique (et reconnaissable) dans l'espace visuel d'un robot. Si ce point a toujours été sur la main du robot, sa cinématique sera celle des mouvements mécaniques du robot. De ce fait, une prédiction pertinente dépendra des positions articulaires, et c'est l'ensemble des articulations qui devra être en entrée. En revanche, si ce point n'avait été qu'artificiellement attaché à la main du robot par l'expérimentateur, celui-ci pourra par la suite l'en retirer pour faire arbitrairement obéir ses mouvements à d'autres lois décorréées de la main du robot.

De fait, en raison de circonstances particulières il est possible que les variations d'un tel point soient, du point de vu du robot, corrélées à des éléments qui peuvent apparaître comme non pertinents pour un observateur extérieur, et qu'il soit de fait difficile d'identifier *a priori* les variables qui seront corrélées avec la sortie observée. Si, par exemple, le robot ne dispose pas des informations relatives aux conditions d'éclairage, mais que celles-ci entraînent une disparition du point observé lorsqu'il est situé dans une zone particulière de l'exploration du robot, on s'attend alors à ce que le robot apprenne une loi spécifique à cette zone et non à la lumière.

D'un autre point de vue, les expériences de substitution sensorielle illustrent le fait que des lois sensorimotrices puissent être apprises à partir de modalités initialement non dédiés à l'usage qui en est fait (par exemple dans les expériences de Bach-y-Rita, un dispositif permet au sens du touché de partiellement se substituer à celui de la vision), et ce par une exploration active de ces lois [Bach-y Rita et al., 1969; O'Regan and Noë, 2001].

Enfin, l'utilisation d'outils illustre également cette idée. En effet, le principe de l'extension du schéma corporel à un outil extérieur au corps suppose bien qu'un ensemble d'éléments sensoriels caractéristiques puisse potentiellement être appris comme modifiant le schéma corporel lui-même. Ainsi, au niveau même de ce qui constitue le corps du robot, c'est-à-dire de ce qui devrait, *a priori*, être le moins variable selon les circonstances (et qui devrait donc pouvoir en pratique, et avec un minimum de risque, être codé directement en dur par le modélisateur), on retrouve une telle plasticité des entrées pertinentes.

En conclusion il semble important, voire nécessaire qu'un robot puisse potentiellement apprendre des lois arbitraires, lesquelles requerrons des entrées potentiellement imprévisibles (notamment dans le nombre de dimensions nécessaires), que ce soit pour le modélisateur ou pour la personne mettant en place les conditions expérimentales.

Propriété des entrées sensorielles

En considérant le point précédent, il nous faut donc prendre en compte, dans leur diversité, les entrées qui sont requises pour apprendre de telles lois. Celles-ci ont les caractéristiques suivantes :

- **Multimodalité** : Afin de prédire les variations d'un senseur, différentes modalités peuvent être sollicitées. Ainsi, les variations de la position de la main pourront dépendre de la proprioception, mais aussi de la vision, par exemple si le robot acquiert la capacité à reconnaître un obstacle.
- **Ajout en cours d'expérience** Dans une perspective développementale, au sens où un robot peut, à l'instar du bébé, développer progressivement de nouvelles capacités motrices et sensorielles (par le relâchement de contraintes maturationnelles, voir par exemple [Law et al., 2014]), de nouveaux degrés de liberté, ou encore des manières plus complexes d'encoder l'information sensorielle ou motrice peuvent se développer. Après une phase d'apprentissage sensorimoteur, l'espace sensorimoteur lui-même peut se trouver agrandi et modifié. Notons ainsi que les évolutions de l'environnement mais aussi de l'appareil sensorimoteur lui-même (le robot) rendent le processus sous-jacent à l'apprentissage sensorimoteur **non stationnaire**.

Propriété principale : indépendance

Considérant ces entrées multimodales, et potentiellement ajoutées en cours d'expérience, examinons maintenant les conséquences que cela peut avoir sur les lois sensorimotrices apprises. Autrement dit, interrogeons-nous sur la possible influence de certains senseurs sur les lois sensorimotrices apprises. Prenons de nouveau un exemple. Imaginons une situation dans laquelle la présence d'un objet à côté de la main d'un robot l'empêche de bouger dans cette direction, et perturbe ainsi l'ensemble des lois sensorimotrices précédemment apprises en l'absence de cet obstacle. Considérons dans cet exemple qu'il y a ici deux modalités permettant d'apprendre des lois sensorimotrices : la première est la proprioception, qui est typiquement utilisée pour prédire les variations de la main, la seconde est la vision. Du point de vue des lois physiques réelles, il existe des règles de dépendances entre des positions articulaires et la position de l'effecteur terminal du robot, autrement dit cette position n'est pas la seule somme des contributions de chaque articulation, mais la contribution de chaque articulation à cette position est directement dépendante des autres positions. En ce sens donc de dépendance, la proprioception et la vision n'ont, eux, pas de dépendances intrinsèques (autres que circonstancielles) : la présence ou l'absence d'objets, ou d'une lumière particulière, n'a pas en soi d'impact sur la position de l'effecteur terminal du robot, même si il peut y avoir un impact globale sur cette position (présence d'un obstacle, ou d'un outil) lié à des circonstances spécifiques.

Ainsi être en un lieu donné, avoir un objet dans la main, voir une lumière particulière, ou encore ressentir une certaine configuration tactile sont des contextes qui peuvent donc potentiellement influencer l'ensemble (ou un sous-ensemble) des lois sensorimotrices d'un robot. Or ces influences globales sont des lois sensorimotrices qui peuvent être apprises, et elles sont à mettre sur le même plan que toutes autres lois sensorimotrices. Ainsi, il semble défendable que "par défaut", l'influence de chaque loi soit globale plutôt que locale. Pour préciser, nous tirerons deux conséquences majeures de cette indépendance entre modalités :

- (i) **Fonctions additives** : Afin de privilégier les effets des entrées indépendantes, nous proposons d'utiliser des fonctions additives globales, pour pouvoir additionner les effets de contextes sensoriels indépendants. L'indépendance, parmi certaines entrées sensorielles, fait qu'il n'y a pas nécessairement d'exclusion mutuelle des effets de ces entrées indépendantes sur la prédiction d'un senseur, et que donc les contributions peuvent s'additionner.

Cette indépendance, traduite en terme de lois additives, est intrinsèquement incrémentale puisqu'il ne s'agit que d'ajouter des lois les unes aux autres lorsque nécessaire, ce qui permet notamment des ajouts à la volée. En effet, considérons qu'un robot apprendre un ensemble de lois en ayant en entrée sensorielle un vecteur θ de n senseurs prédisant correctement les variations d'un senseur V donné par la fonction $f_{1..n}$. Supposons que ces prédictions sont correctes. Considérons maintenant la présence d'un nouveau senseur θ' (pour l'une des raisons citées précédemment, comme par exemple la libération, au sein d'un processus développemental, d'un degré de liberté). L'usage de fonctions additives revient à ne devoir apprendre ici qu'une unique nouvelle loi, $f_{n+1}(\theta')$, puisque l'on pourra appliquer la formule $V = f_{1..n}(\theta) + f_{n+1}(\theta')$. L'idée, comme nous l'avons dit, est que la majeure partie des lois sensorimotrices qui nous intéressent (lois multimodales, et/ou enrichies incrémentalement) auront cette propriété d'additivité, et il nous importera donc d'avoir un mécanisme de recrutement robuste à cette caractéristique.

- (ii) **Frontière d'activation** : Reprenons le dernier exemple : avant l'activation du nouveau degré de liberté θ' , ayant entraîné une erreur, les prédictions données par $f_{1..n}(\theta)$ étaient correctes. Ainsi, la découverte de la nouvelle loi est ici liée à la variation de θ' (un senseur dont la variation θ' est le moteur associé). De fait la découverte d'une nouvelle loi consiste aussi, pour un robot parcourant son espace sensorimoteur, en la découverte d'une frontière entre le domaine où cette loi s'applique (soit celui où le robot se situe au moment de la découverte de cette nouvelle loi), et le domaine où elle ne s'applique pas (soit celui où le robot effectuait de bonnes prédictions). Dans notre exemple, puisque la variation de θ' a entraîné la découverte d'une nouvelle loi (les précédentes échouants), on peut supposer que cette nouvelle loi dépend uniquement de la valeur de cette variable, et que la loi d'activation déterminant l'usage de la nouvelle loi sensorimotrice devrait définir une frontière entre deux domaines (que l'on pourrait qualifier de globaux car séparant l'espace en deux) plutôt que délimiter une zone selon une structure prédéfinie paramétrable (et donc locale, comme une gaussienne par exemple).

Soulignons dès maintenant que ce que nous proposons dans ce chapitre est d'examiner un modèle qui privilégie la propriété d'indépendance, et plus précisément en le basant sur les deux points que nous venons de mentionner. Si l'idée est bien de rendre compte au mieux de l'aspect incrémental et multimodal du problème qui nous occupe et qui se manifeste fréquemment lors de passage à des échelles supérieures (ajout de senseurs ou de modalités), et d'être efficient en ces cas, nos expériences se concentreront au contraire sur les résultats obtenus lors de situations de fortes "dépendances" afin de montrer la validité et l'efficacité du modèle même dans ces cas a priori les plus inadaptés, mais qui peuvent précéder ces passages à l'échelle.

Afin de proposer un tel modèle, nous allons donc nous intéresser aux limitations de l'actuel DSM, notamment au vu de ce qui a été dit.

4.1.2 Conséquences et problèmes de l'actuelle catégorisation

Dans cette partie, nous allons faire un examen critique du système de recrutement de catégories (le SAW) décrit dans la partie 3, en faisant le lien avec les deux propriétés (i) et (ii), concernant l'usage de fonctions additives (i), lesquels devraient avoir une loi d'activation caractérisant une frontières entre deux espaces (ii), comme nous l'avons décrits précédemment. Précisons ici qu'une catégorie correspond à l'encodage d'un contexte, et que, comme nous l'avons vu en introduction de ce chapitre, c'est cette notion que nous allons faire évoluer.

Rappelons que dans le SLE, lorsqu'une erreur est découverte, et donc qu'une nouvelle loi doit être créée, il y a 3 étapes successives :

- 1 Le SAW recrute une nouvelle catégorie correspondant aux valeurs courantes des entrées sensorielles au moment du recrutement
- 2 Le WTA sélectionne la catégorie la plus activée, la met à 1 et les autres à 0
- 3 Le LMS apprend à associer cette catégorie à la loi linéaire reliant un moteur à un senseur.

Dans un premier temps (section 4.1.2) nous nous intéresserons à ce qui est en jeu à l'étape 1, en faisant le lien avec la propriété (ii). Puis (section 4.1.2), nous nous intéresserons à ce qui est en jeu dans les étapes 2 et 3, en faisant cette fois le lien avec la propriété (i).

Limites liées au recrutement et au partitionnement de l'espace

Distance commune : Lorsqu'une nouvelle loi est créée, celle-ci est associée à une nouvelle catégorie elle aussi créée, laquelle est reliée à tous les senseurs en entrée avec un coefficient égal à la valeur des senseurs à ce moment là (voir sec. 3.1.1 pour le détail des notations) : $\forall j, w_{kj}^{SAW^{na}} \leftarrow s_j^{Real}$. La loi d'activation de cette catégorie est alors $A_k = 1 - \frac{1}{N} \sum_j |w_{kj} - e_j|$.

Ce cas illustre celui, plus général, de lois d'activation basées sur un voisinage, dans lequel l'activation de la catégorie dépend globalement d'une distance à la configuration des senseurs au moment du recrutement (même si, dans certains cas, le centroïde peut se déplacer). La distance utilisée est généralement définie comme étant initialement la même pour tous les senseurs, et donc chaque senseur est a priori également pris en compte (même si les distances peuvent aussi être paramétrables, ou évoluer au cours de l'apprentissage).

Or, une mesure de distance commune à différents senseurs suppose implicitement que ceux-ci soient commensurables, c'est-à-dire qu'il y ait du sens à ce que les différentes grandeurs soient comparables. Mais, au vu des propriétés des entrées, décrites précédemment, supposer des ordres de grandeurs comparables pose problème, dans la mesure où ces grandeurs dépendent des dimensions propres de l'espace encodé (à commencer par le nombre de dimensions d'une modalité pas exemple, ou l'importance de chacune pour la loi à encoder), et de l'encodage lui-même (de la métrique utilisée, par exemple en norme 1 ou 2 dans un espace de plusieurs dimensions, et surtout, de l'unité qui est utilisée pour les différentes dimensions, comme les mètres, pixels, degrés, etc.).

Enfin, rappelons que cette métrique sert à l'activation de contextes, et, par extension, sert à l'activation des lois qui sont associées à ces contextes. Il semblerait donc logique que cette métrique soit inférée de la distance entre les lois (notion de frontière entre deux lois dont nous avons parlé plus haut), plutôt que d'une définition *a priori* de cette métrique, quand bien même celle-ci peut-être adaptée par l'expérimentation, en ligne ou par le modélisateur lui-même.

Explosion combinatoire : Dans le cadre actuel, le SAW peut donner lieu à une explosion combinatoire, et la “malédiction” de la dimensionnalité est ici décuplée. En effet, la catégorisation à l’aide d’un SAW est dans notre cas non seulement exponentielle, mais aussi moins efficace et pertinente à mesure que le nombre d’entrées augmente. Dans le pire cas, si par exemple un seul senseur permet de catégoriser efficacement la variation d’une loi, alors puisque c’est toujours, malgré tout, le vecteur de senseurs entier qui est catégorisé, c’est toute la combinatoire avec les autres senseurs qui doit être apprise, ce qui ajoute autant de catégories non pertinentes. Ce cas est illustré dans la section 2.1, où suite à l’ajout d’un outil lors d’un apprentissage visio-moteur, l’information tactile (qui indique une présence dans la main) devait seule prédire une certaine variation de la position de la main, mais c’est l’ensemble de la combinatoire liée à la proprioception qui a dû de nouveau être appris.

Généralisation pour les séquences : Comme nous l’avons souligné dans le chapitre 3, la limitation principale de DSM se situe selon nous dans le fait que les entrées pertinentes sont, dans les expériences que nous avons effectuées, choisies par la personne faisant le modèle. Ces entrées sont choisies à la fois pour leur influence sur la sortie prédite, afin que le prédicteur (SLE) ait les éléments lui permettant de catégoriser correctement les entrées sensorielles, mais elles seront aussi utilisées, par la suite, pour effectuer des séquences d’actions. En effet, dans les expériences présentées à la section 3.2.4, une recherche sur les entrées possibles de S est effectuée par la simulation, et consiste à obtenir un S^{Sim} permettant d’avoir une forte valeur d’affordance, en générant pour cela plusieurs configurations aléatoires sur l’ensemble des entrées sensorielles. Ceci permet de trouver une configuration particulière (de S , donnée par S^{Sim}) à même de débloquer une situation et de satisfaire une tâche. Pour cela, le modèle va envoyer des valeurs de senseurs ‘désirées’, ce qui rendra possible la séquence d’action permettant d’atteindre le but désiré.

Ainsi, dans les expériences (et le modèle) présentées jusqu’à maintenant, la configuration recherchée, et qui sera par la suite désirée, concerne l’ensemble des senseurs en entrée. Or, si, comme dans l’exemple précédent, une seule de ces entrées est véritablement nécessaire pour se mettre dans un contexte réel d’application d’une loi désirée, ce sera malgré tout toujours l’ensemble des senseurs qui devront systématiquement prendre une configuration particulière. Ces configurations pourront donc comporter des changements non nécessaires, et même potentiellement contraignants, nuisant de fait à la généralisation de l’obtention de ce résultat, dans la mesure où le résultat désiré aura tendance à ne pouvoir être obtenu que dans des configurations sensorielles pratiquement similaires à celles déjà rencontrées.

Limites liées à l’additivité ou à la non additivité

Lors du calcul de la loi sensorimotrice à appliquer, deux solutions opposées sont potentiellement possibles : soit l’ensemble des catégories, et donc des lois sensorimotrices associées est prise en compte, soit seulement une unique catégorie est prise en compte, avec sa loi associée, ce qui est là le cas du SLE (voir sec. 3.1.1) avec la présence du WTA en sortie du SAW. De fait, nous allons ici discuter des effets de la présence ou de l’absence du WTA en sortie du SAW. Bien que des solutions intermédiaires soient possibles (comme par exemple dans l’algorithme RFWR, voir [Schaal and Atkeson, 1998], où la fonction d’apprentissage définit un certain voisinage, typiquement gaussien, et est également utilisée pour pondérer les lois linéaires associées), nous

nous intéresserons à ces deux situations opposées afin de pointer des désavantages propres à chacun de ces extrêmes.

- En présence d'un WTA (comme dans le cas que nous avons présenté dans la partie 3), une unique catégorie est considérée, la gagnante étant à 1 et les autres à 0. Ainsi cette catégorie gagnante l'est à l'exclusion de toutes les autres catégories, dont les lois associées ne seront alors pas exploitées. Ceci entraîne une forme d'oubli, au moment d'un nouvel apprentissage, des apprentissages passés, en ne permettant chaque fois qu'un apprentissage de lois applicables dans des configurations précises et leur voisinage, et à l'exclusion des autres régions de cet espace (lequel, dans nos expérimentations, aura été prédéfini). Les anciennes lois ne sont donc pas généralisées, ni utilisées pour construire les nouvelles, et chaque nouvelle loi ne bénéficie pas de ce qui avait été précédemment appris. De la sorte, l'indépendance évoquée précédemment, qui peut notamment être liée à la multimodalité, ne pourra pas être exploitée au sens où une loi relative à un sous ensemble des senseurs (comme par exemple la proprioception) ne pourra pas être ajoutée aux lois relatives à d'autres senseurs indépendants (comme le tactile). Ainsi dans notre exemple de la section 2.1, on ne pourrait pas, par la méthode que nous venons de décrire, ajouter la contribution de l'outil au précédent apprentissage.
- Si, à l'inverse, chacune des catégories est potentiellement activée selon une fonction donnée (comme typiquement une gaussienne), alors au contraire toutes les autres lois seront théoriquement en mesure de participer dans n'importe quelle situation (selon la fonction choisie). En effet, en ce cas les lois associées à chaque catégorie dont l'activation est non nulle s'additionneront entre elles (dans notre modèle, avec le LMS qui suit). Or, de cette manière, au moment du recrutement d'une nouvelle catégorie, celle-ci pourra être associée à une loi sensorimotrice dépendante de toutes les autres lois qui se trouvent être activées à cet instant. Toutefois, les lois associées aux catégories précédemment recrutées auront, elles, été apprises en l'absence de cette nouvelle catégorie. De fait, lorsque le robot se trouvera être dans une situation similaire à l'une de celles qu'il aura déjà traversées, la nouvelle catégorie pourra potentiellement être elle aussi activée, et sa loi associée pourra de fait venir s'additionner, et donc interférer, avec les autres lois. Il y aura de ce fait une forme de désapprentissage (ou "oubli catastrophique", voir [McCloskey and Cohen, 1989; Ratcliff, 1990]), qui, au mieux, nécessitera une réadaptation de l'ensemble des anciens apprentissages.

4.1.3 Proposition

Notre proposition est donc de considérer avant tout ces entrées comme fondamentalement indépendantes (qu'elles le soient de fait ou non), et ainsi de proposer une solution aux propriétés (i) et (ii), concernant l'usage de fonctions additives (i), lesquels devraient avoir une loi d'activation caractérisant une frontière entre deux espaces (ii), décrites dans la section 4.1.1.

Au sein de DSM, le recrutement d'une nouvelle catégorie s'accompagne :

- d'une loi d'activation, basée sur les senseurs en entrée au moment du recrutement, et qui utilise une mesure de distance
- d'une loi sensorimotrice associée, qui est le rapport linéaire entre la magnitude motrice et la variation d'un senseur observé également au moment du recrutement.

Jusqu'alors, la création d'une nouvelle catégorie était basée sur les données disponibles au moment du recrutement uniquement, et la nouveauté d'une catégorie était :

- donnée par une mesure de distance indépendante des autres catégories, mais dépendante des distances arbitraires entre les différents senseurs
- indépendante à ce que la loi qui lui était associée apportait justement de nouveau, en ce que l'apprentissage ne tenait pas compte des lois pré-existantes pour être apprise

Le principe que nous voulons donc ici mettre en avant est qu'une nouvelle catégorie n'est nécessaire qu'au regard des connaissances actuelles du robot, et donc par rapport aux connaissances déjà accumulées. Ainsi, nous proposons que la nouveauté de la catégorie soit donnée relativement aux autres, et qu'elle n'existe ainsi que de par sa relation, et plus précisément par son opposition avec les catégories qui lui sont associées (pour un couple sensorimoteur donné). On voudra ainsi définir les catégories de manière purement différentielle, non pas positivement par leur contenu, mais négativement par leurs rapports avec les autres catégories.

Ainsi, nous proposons :

- (i) D'associer une nouvelle catégorie à une loi sensorimotrice qui soit non absolue, mais relative aux lois existantes (c'est-à-dire relative à celles déjà apprises, et qui ont permis d'obtenir une prédiction donnée Δs^{Pred}) : la nouvelle loi apprendra alors l'erreur ($\Delta s^{Pred} - \Delta s^{Real}$), et non la loi courante constatée (Δs^{Real}). Afin d'obtenir une prédiction correcte, le **principe d'additivité** sera donc pris en compte. Une nouvelle loi sera incrémentée aux précédentes, par l'ajout d'une catégorie dont les effets viendront s'additionner à l'ensemble des effets des précédentes catégories.
- (ii) De créer des lois d'activations propres à chaque catégorie, et relatives aux anciennes catégories. Afin d'éviter que l'activation d'une nouvelle catégorie, dans les situations précédemment rencontrées, ne soit susceptible de faire échouer des prédictions jusqu'alors correctes, par l'ajout un correctif de trop, la règle d'activation propre à cette nouvelle catégorie sera définie de telle sorte que l'activité soit nulle dans les situations antérieures, et activée (c'est-à-dire égale à 1) au moment où la catégorie est recrutée. Cette loi d'activation, définie négativement, révélera ainsi la **frontière d'activation** de la loi additive apprise (cette loi d'activation sera alors globale et non locale).

Notons que le but du robot sera donc d'apprendre simultanément deux sortes de lois sensorimotrices : l'une concerne les rapports entre moteurs et senseurs, et l'autre concerne les rapports entre d'une part la variation des senseurs (apprise grâce aux lois d'activation spécifiques à chaque catégorie) et d'autre part une variation du rapport entre moteur et senseur (qui se traduira ici par une addition). Chacune de ces règles aura été expérimentée par le robot, au sens où les moteurs auront été réellement activés, tout comme la variation des senseurs aura été constatée par le robot (afin de créer la loi d'activation). Précisons qu'en faisant ceci, nous supposons de fait qu'un robot qui se trouve parcourir son espace des senseurs, en étant situé à un endroit donné, ne sera pas sujet à des "sauts" spatio-temporels au sein de cet espace. Autrement dit, même avec une fréquence d'échantillonnage faible, l'espace ne devra pas être l'objet d'une exploration stochastique mais devra au contraire être parcouru de manière continue, dans le temps et dans l'espace.

Dans la lignée de notre approche sensorimotrice, c'est précisément ce parcours qui rendra le robot à même de saisir l'évolution des lois de co-variations entre l'environnement et ses actions, et ainsi de saisir l'ensemble de ses lois sensorimotrices lui permettant à la fois de savoir comment agir, et comment agir sur ces lois.

Nous proposons ainsi que l'une des spécificités d'un robot, dans le domaine de l'apprentissage automatique, soit de justement appréhender de manière continue dans l'espace et le temps, l'espace des senseurs, grâce au fait que le robot y soit situé. Ceci nous permet ainsi de définir ce qu'est un lieu, dans cet espace, non pas positivement par rapport aux caractéristiques absolues de ce lieu, mais de le définir uniquement négativement, relativement aux autres lieux parcourus, en ne faisant que le différencier des lieux, et des lois qui y sont associées, qui auront été précédemment parcourus et identifiés.

L'hypothèse sous jacente à notre proposition est que la continuité temporelle permet de garantir la perception de la causalité des événements (même si des "états cachés", c'est-à-dire non observables et donc non perceptibles par le robot mais influençant les événements ou lois, peuvent bien sûr exister). Notre modèle, dans la version que nous présenterons, ignorera ainsi de tels états cachés (qu'il faudrait intégrer par la suite) et postulera donc implicitement que toute causalité pourra être saisie par l'exploration sensorimotrice de l'environnement. De fait, si une loi sensorimotrice varie (ce qui se traduit par une erreur de prédiction), alors la cause de cette variation devrait pouvoir être saisie par l'observation de la variation des valeurs de senseurs.

Cependant, même si la cause est saisissable au sein de la variation des valeurs de senseurs, et est donc potentiellement accessible au robot grâce à l'observation de ces variations, elle ne réside pas nécessairement dans l'ensemble des éléments ayant varié. Pour illustrer cela, prenons l'exemple d'un robot humanoïde doté d'un bras. Nous pouvons prédire la variation de la position de la main, lorsque le bras est tendu et que l'épaule bouge. Si, après cela, le même mouvement est effectué, mais alors que le coude est plié, alors la loi de covariation entre la position de la main et les mouvements de l'épaule pourra se trouver changée, par exemple en entraînant une amplitude de variations moindre de la position de la main lors de mouvements. Dans le scénario où, après avoir appris la loi entre l'épaule et la position de la main lorsque le bras est tendu, le robot découvre une nouvelle loi lorsque le coude est plié, alors la variation de la loi est fonction de la variation de la configuration sensorielle, et cette fonction, appelée f , apprenant la variation de la configuration sensorielle sera fonction de la variation de l'angle du coude. Dès lors, la fonction f ("angle du coude") sera apprise pour prédire le changement de la loi. Mais supposons maintenant qu'en même temps que le coude se replie, le poignet effectue lui un mouvement de rotation. La différence de configurations sensorielles sera une fonction du type f ("angle du coude ET rotation du poignet"), bien que le mouvement du poignet n'ait aucune influence sur la loi en question.

Il apparaît alors que si le changement de configuration sensorielle peut contenir l'information de causalité, ces changements peuvent toutefois englober plus d'éléments que ceux causant le changement de lois. En conséquence, cette cause devra faire l'objet d'une recherche au sein du sous-ensemble donné par ces variations de senseurs.

Nous allons dans la section qui suit présenter un modèle permettant d'implémenter ces idées, et de les intégrer à DSM.

4.2 Modèle pour la régression dynamique de lois sensorimotrices

4.2.1 Principe

Le changement de configuration sensorielle, et par voie de conséquence, la causalité, apparaît sous la forme d'une conjonction $\Delta^C = \delta s_i \wedge \delta s_j \wedge \dots$, avec s_i et s_j des senseurs. Dans notre modèle "Sensorimotor Law Regressor" (SLR), nous proposons qu'initialement, une nouvelle catégorie soit comme précédemment recrutée en cas d'erreur (détectée par un SLE, *Sensorimotor Law Encoder*), mais ait ici, comme loi d'activation, la conjonction des lois relatives de chaque variation, que nous allons décrire.

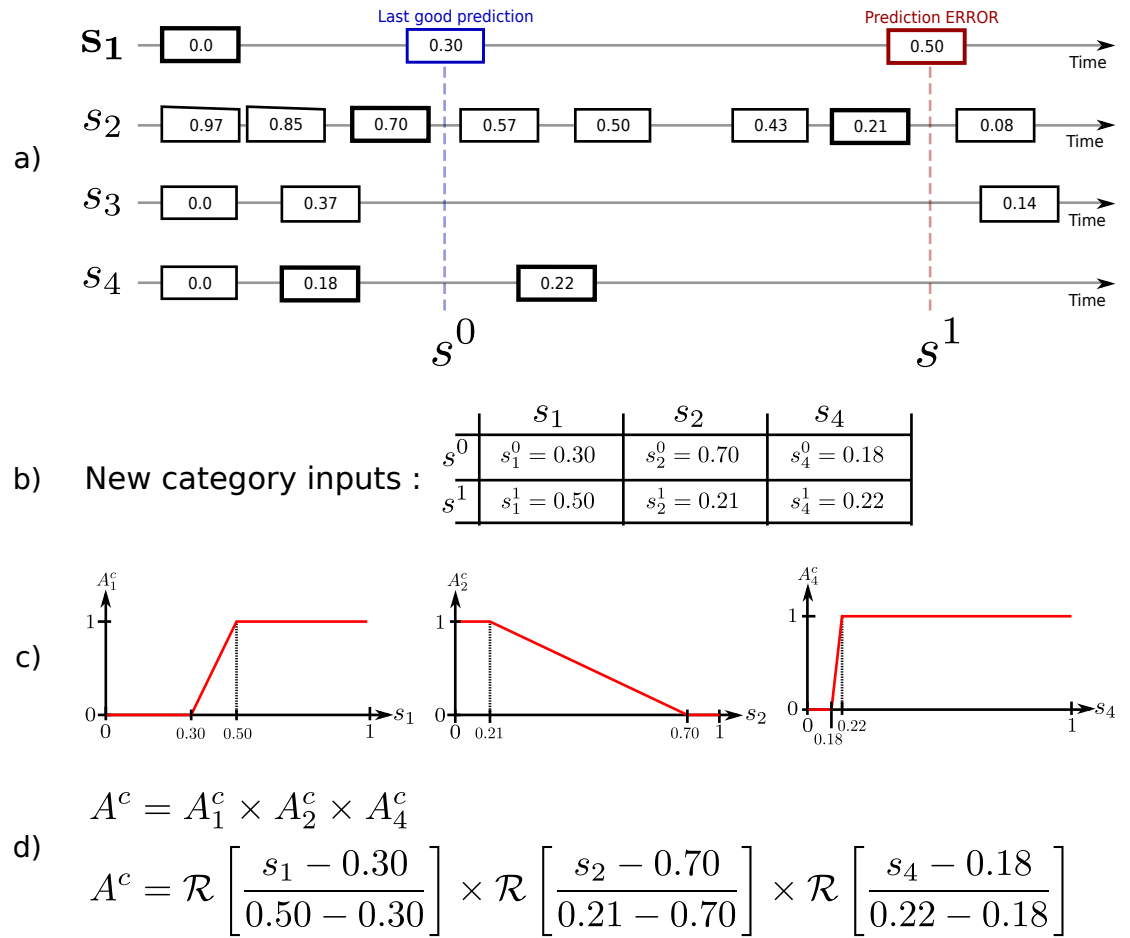


FIGURE 4.1 – Exemple de création de loi d'activation lors du recrutement d'une catégorie, en l'occurrence suite à une erreur constatée sur le senseur s_1 . En a) un exemple de valeurs de 4 senseurs au cours du temps, b) les valeurs des 3 senseurs ayant suffisamment varié depuis la précédente bonne prédiction de s_1 . En c) les 3 rampes correspondantes utilisées pour la loi d'activation. En d), la loi d'activation correspondante.

Prenons le cas où une erreur apparaît sur le senseur j (donc détectée par le SLE_j). Alors, en appelant t l'instant où le SLE_j a effectué la dernière prédiction correcte $E_{rr}^{SLE_j}(t) = 0$ (c'est-à-dire un moment où aucune erreur n'a été détectée lorsque la vérification est faite, lorsque

$|\Delta s_i^{Pred} - \Delta s_i^{Real}| > Th_1^{Err}$, voir ligne 5 de l'Algorithme 1) et en appelant t' l'instant où une erreur est détectée par le SLE $E_{rr}^{SLE_j}(t') = 1$, alors les variations de senseurs Δ^C qui seront considérées seront celles qui auront lieu entre t et t' . Supposons maintenant que plusieurs senseurs aient varié, dont par exemple les senseurs i et k : $\Delta^C = \delta s_i \wedge \dots \wedge \delta s_k$. Comme nous l'avons dit plus haut, la loi d'activation de la catégorie recrutée au moment de l'erreur sera la conjonction des contributions de chaque variation de senseurs, sur l'intervalle de temps que nous avons défini. Notons s_i^0 la valeur du senseur i au moment t , c'est-à-dire au moment où la prédiction était encore bonne, et s_i^1 la valeur du senseur i au moment t' , soit sa valeur lorsque la prédiction a échoué. En notant A^c la loi d'activation de la catégorie c , alors la contribution, notée A_i^c , de la variation de s_i à la loi d'activation est la suivante :

$$A_i^c = \mathcal{R} \left[\frac{s_i - s_i^0}{s_i^1 - s_i^0} \right] \quad (4.1)$$

Avec

$$\mathcal{R}[x] = \begin{cases} 1 & \text{si } x > 1 \\ 0 & \text{si } x < 0 \\ x & \text{sinon.} \end{cases} \quad (4.2)$$

Il s'agit donc d'une rampe définissant la frontière entre les valeurs de s_i pour lesquelles la catégorie sera activée (soit à la valeur 1) et celles pour lesquelles cette catégorie sera désactivée (soit à la valeur 0). Notons qu'une fonction de type sigmoïde pourrait ici être envisagée.

Pour une variation de senseurs de type $\Delta^C = \delta s_i \wedge \dots \wedge \delta s_k$ entre une prédiction correcte et une erreur, la loi d'activation totale est alors la conjonction de ces contributions, et est donc de la forme :

$$A^c = A_i^c \times \dots \times A_k^c \quad (4.3)$$

Pour illustrer ceci, prenons un exemple (représenté figure 4.1). Supposons que l'on souhaite prédire les variations du senseur s_1 (via la SLE_1). Sur la figure 4.1-a), les 4 lignes représentent l'évolution de 4 senseurs en entrée, et les rectangles sont des exemples de leurs valeurs au cours du temps. Supposons donc qu'une erreur de prédiction soit constatée par le SLE_1 . Alors la nouvelle catégorie apprendra la correction qu'il faudra additionner à la prédiction qui a échoué, et aura comme entrée, déterminant effectivement sa loi d'activation, l'ensemble des senseurs ayant "suffisamment" varié (nous avons, par simplicité, choisi d'utiliser pour cela le seuil Th_1^{Err}) depuis la dernière bonne prédiction de ce SLE. À la dernière bonne prédiction donc, les valeurs de chaque senseur sont enregistrées dans un vecteur noté s^0 . Ceci permet par la suite de déterminer parmi ces senseurs en entrée ceux ayant suffisamment varié, et de créer la loi d'activation. Dans notre exemple, où c'est la loi de s_1 qui doit être prédite, cette loi sera ainsi basée sur les senseurs s_1 , s_2 et s_4 (voir fig.4.1-b)) puisqu'on peut voir en a) qu'eux seuls ont varié suffisamment entre la dernière bonne prédiction, et l'erreur de prédiction. À partir de chaque valeur s^0 et s^1 provenant des valeurs de ces senseurs au moment de l'erreur, la loi d'activation de la nouvelle catégorie est créée (fig.4.1-d), en créant les 3 rampes correspondantes (fig.4.1-c).

Il faut noter le cas particulier d'un tout premier mouvement (qui se présente en début d'expérience), dans lequel il n'y a pas de passé et donc où il ne peut y avoir de précédente prédiction correcte. Afin de pallier ce cas particulier (qui n'arrive qu'une fois par SLE), deux possibilités sont envisageables :

- Soit lors du début de l'expérience toutes les prédictions sont considérées comme correctes. Ce qui revient en fait à considérer le cas particulier où, dans la configuration sensorielle initial du robot, aucune activité motrice ne générerait de variation de capteur. Ainsi, ces premières lois seraient presque systématiquement fausses, ce qui nécessiterait d'autres recrutements pour les corriger (résultats confirmés par des expérimentations non reportées ici).
- Soit on considère que la première loi est toujours valide, en donnant comme activation à chaque première catégorie recrutée : $A^c = 1$. Cette option, que nous avons choisie, respecte l'idée générale que nous avons mentionnée, selon laquelle chaque loi doit être généralisée partout où elle n'affecte pas de précédents apprentissages (ce qui est ici le cas car il n'y a pas ici de précédents apprentissages).

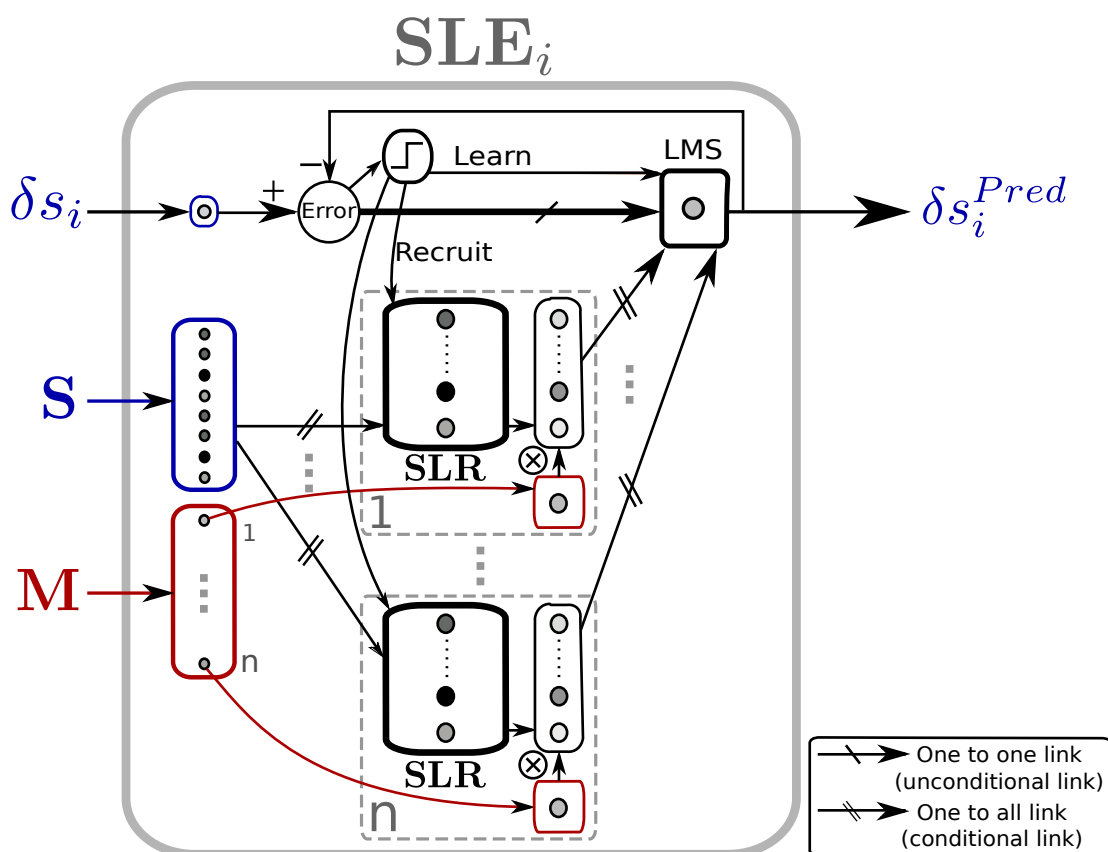


FIGURE 4.2 – Nouvelle version du SLE. Les changements par rapport à la précédente version (fig.3.1) sont en gras : le SLR vient remplacer le SAW et le WTA, et le LMS a désormais l'erreur de prédiction en entrée conditionnelle (et non directement la valeur à prédire).

4.2.2 Fonctionnement, et intégration à DSM

Tout d'abord le SLE est modifié pour substituer au SAW, et au WTA, le SLR que nous venons de décrire, ce que représente la figure 4.2.

Le SLR est toujours suivi d'une multiplication par la magnitude motrice. Le LMS qui succède au SLR reste inchangé, toutefois, ses 2 entrées sont différentes. D'une part, comme il n'est plus précédé d'un WTA, c'est l'ensemble des catégories, multiplié par la magnitude motrice associée, qui additionneront leurs contributions. D'autre part, chaque poids appris par le LMS n'est plus la variation prédite courante Δs_i^{Real} , comme c'était le cas précédemment (voir sec. 3.1.1), mais est désormais l'erreur, c'est-à-dire la différence entre la variation réelle et celle prédite soit $\Delta s^{Pred} - \Delta s^{Real}$. De la sorte, la prédiction exacte sera obtenue par l'addition des précédentes contributions à la nouvelle.

Dans l'Algorithme 2, nous montrons en quoi les équations diffèrent de celles de l'algorithme précédent (Algorithme 1). Les seules différences sont aux initialisations qui sont faites lors d'un recrutement, ici aux lignes 7 pour le SLR et 8 pour le LMS, ainsi que pour le calcul de la loi d'activation du SLR, ligne 15.

Algorithm 2 : $SLE_i^{SLR}(i, \mathbf{S}^{Real}, \mathbf{M}^{Real})$

```

1:  $s_i^{Stored} \leftarrow s_i^{Real}$  ;  $\Delta s_i^{Pred} \leftarrow 0$  ;  $\forall n, M_n \leftarrow 0$ 
2: loop
3:    $\Delta s_i^{Real} \leftarrow s_i^{Real} - s_i^{Stored}$ 
4:   if  $|\Delta s_i^{Real}| > Th_1^{Step}$  or  $|\Delta s_i^{Pred}| > Th_1^{Step}$  then
5:     if  $|\Delta s_i^{Pred} - \Delta s_i^{Real}| > Th_1^{Err}$  then
6:       // Error : Recruitment for the (unique) activated motor  $na$ , i.e. where  $M_n \neq 0$ , of the  $k^{th}$ 
       // neuron in  $SLR^{na}$ , and learning in  $LMS^{na}$ 
7:       for all input sensors  $j$  that have moved since last good prediction, set  $j_1^0, j_1^1, \dots, j_2^0, j_2^1$ 
8:          $w_k^{LMS^{na}} \leftarrow \frac{\Delta s^{Pred} - \Delta s^{Real}}{M_{na}}$ 
9:       end if
10:      // New step : re-initialisation of the predictor
11:       $s_i^{Stored} \leftarrow s_i^{Real}$  ;  $\Delta s_i^{Pred} \leftarrow 0$  ;  $\forall n, M_n \leftarrow 0$ 
12:    end if
13:    // Perform current prediction
14:     $\forall n, M_n \leftarrow M_n + m_n$ 
15:     $\forall n, \left\{ \begin{array}{l} \forall k, A_k^{n,k} \leftarrow \mathcal{R} \left[ \frac{s_{j_1} - s_{j_1}^0}{s_{j_1}^1 - s_{j_1}^0} \right] \times \dots \times \mathcal{R} \left[ \frac{s_{j_2} - s_{j_2}^0}{s_{j_2}^1 - s_{j_2}^0} \right] \\ \forall k, B_k^n \leftarrow m_n A_k^n \end{array} \right.$ 
16:     $\delta s_i^{Pred} \leftarrow \sum_n \sum_k w_{kn}^{LMS^n} B_k^n$ 
17:     $\Delta s_i^{Pred} \leftarrow \Delta s_i^{Pred} + \delta s_i^{Pred}$ 
18:  end loop

```

Mais ce modèle, en l'état, est sujet au problème évoqué à la section précédente (sec. 4.1.3) : si un changement de configuration sensorielle peut contenir l'information sur la causalité, permettant de définir une frontière dans l'application de lois sensorimotrices, cette causalité n'est pas nécessairement liée à l'ensemble des senseurs qui se trouvent avoir varié. Certains de ces senseurs peuvent n'avoir aucune influence sur le changement de loi constaté au moment d'une erreur. On peut dès lors imaginer de multiples manières de sélectionner un sous-ensemble de ces variations sensorielles (en n'en faisant aucune, ou par un calcul statistique permettant de

lier chaque cause aux conséquences, voire les causes entre elles en regardant les covariations de senseurs, etc.). (Notons que sans aucune sélection, et donc avec le modèle en l'état, on pourrait douter de la capacité du modèle à converger, et des expériences préliminaires, non reportées ici, vont en ce sens.)

Dans la lignée de notre approche sensorimotrice, nous proposons que cette sélection se fasse par l'expérimentation, et plus précisément par des actions du robot, lui permettant de lever cette ambiguïté. Si l'on suppose que l'enfant est capable d'actions lui permettant d'évaluer les causes de ses prédictions, et que ces actions peuvent en dernier lieu être atomisées en sous actions spécifiques, il semble également permis de supposer que de telles actions puissent servir à affiner les prédictions effectuées, et qu'ainsi une partie de l'activité du robot soit motivée non par un but extrinsèque mais par la motivation d'affiner ses propres connaissances (voir section 3.2 ainsi que la question des motivations intrinsèques, et plus précisément du comportement de jeu, dans l'introduction de ce chapitre 4).

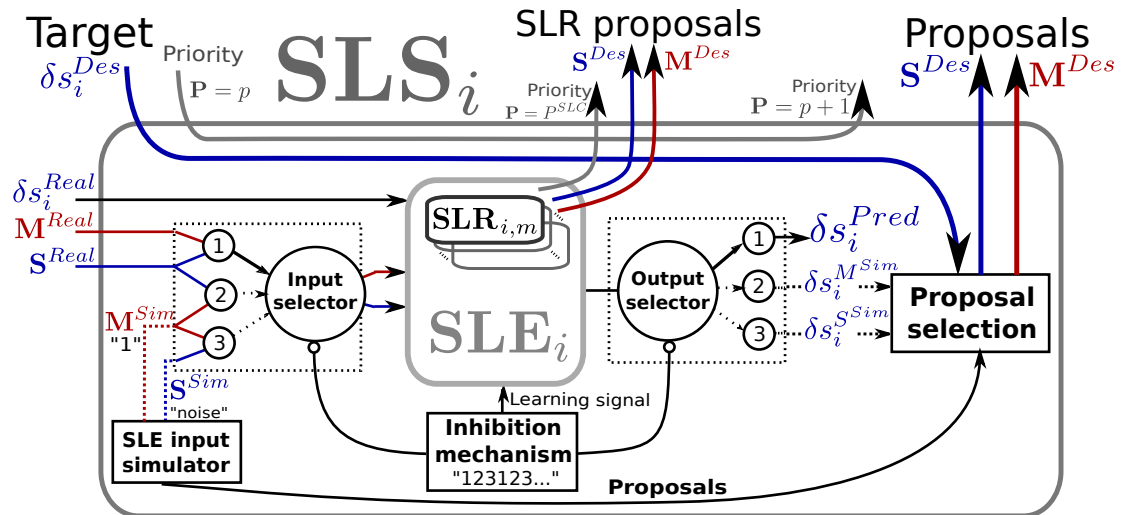


FIGURE 4.3 – Nouvelle version du SLS, avec le SLR. La seule différence avec la précédente version du SLS, fig.3.8, est la présence de nouvelles sorties associées au SLR, mais de même type (S^{Des} , M^{Des} et la priorité P^{SLR} associée).

Pour tester les causes, tant par l'action que par une série d'actions (permettant de tester le "moteur" du changement de loi dont nous parlons en introduction), nous proposons que le SLR puisse, au même titre qu'un SLS (Simulateur de Lois Sensorimotrices, voir sec. 3.2.2), émettre des configurations de senseurs désirées (ou sous-buts) S^{Des} , et des activités motrices désirées M^{Des} . Or chaque SLR est inclus dans un SLE (voir fig.4.2), lequel est lui-même inclus dans un SLS (voir fig.3.8). De fait, l'ajout de cette fonctionnalité au sein de DSM est immédiate : le SLR ayant des sorties similaires aux SLS, celles-ci en suivront le même chemin. Nous pouvons voir sur la figure 4.3 la nouvelle version du SLS. On y voit un nombre de sorties plus important que précédemment, mais celles-ci sont toujours du même type, et elles seront comme précédemment en compétition dans DSM (la figure du schéma global de DSM, fig.3.9, reste inchangée). De fait, rien d'autre n'est changé, il y aura seulement, durant les phases de recrutement, plus de propositions de S^{Des} et de M^{Des} . Pour ce qui est de la priorité associée à ces demandes, les

sorties du SLR auront une priorité constante P^{SLR} (que nous avons choisie élevée afin d'en faire une tâche prioritaire).

Afin d'intégrer le comportement de jeu, résultant des actions et sous-buts désirés par les SLRs, l'algorithme du SLE trouve sa version définitive dans l'Algorithme 3. La seule différence avec la version précédente (l'Algorithme 2) est que le recrutement, ainsi que les différentes étapes de jeu qui suivent, ont été regroupés dans une fonction **SLR** dont nous allons également décrire l'algorithme.

Algorithm 3 : SLE $_i^{SLR,Play}$ ($i, \mathbf{S}^{Real}, \mathbf{M}^{Real}$)

```

1:  $s_i^{Stored} \leftarrow s_i^{Real}$  ;  $\Delta s_i^{Pred} \leftarrow 0$  ;  $\forall n, M_n \leftarrow 0$ 
2: loop
3:    $\Delta s_i^{Real} \leftarrow s_i^{Real} - s_i^{Stored}$ 
4:    $E_{rr}^{SLE_i} = 0$ 
5:   if  $|\Delta s_i^{Real}| > Th_1^{Step}$  or  $|\Delta s_i^{Pred}| > Th_1^{Step}$  then
6:     if  $|\Delta s_i^{Pred} - \Delta s_i^{Real}| > Th_1^{Err}$  then
7:       // Error : Recruitment for the (unique) activated motor  $na$ , i.e. where  $M_n \neq 0$ , of the  $k^{th}$ 
       // neuron in  $SLR^{na}$ , and learning in  $LM S^{na}$ 
8:        $E_{rr}^{SLE_i} = 1$ ,  $\forall n$ , if  $n = na$  then  $is\_moving^{na} \leftarrow 1$ , else  $is\_moving^{na} \leftarrow 0$ 
9:        $w_k^{LM S^{na}} \leftarrow \frac{\Delta s^{Pred} - \Delta s^{Real}}{M_{na}}$ 
10:      else
11:        // Prediction OK
12:         $E_{rr}^{SLE_i} = -1$ 
13:      end if
14:      // New step : re-initialisation of the predictor
15:       $s_i^{Stored} \leftarrow s_i^{Real}$  ;  $\Delta s_i^{Pred} \leftarrow 0$  ;  $\forall n, M_n \leftarrow 0$ 
16:    end if
17:    // Perform current prediction
18:     $\forall n, M_n \leftarrow M_n + m_n$ 
19:     $\{\mathbf{A}, \mathbf{S}^{Des}, move_m^{na}\} \leftarrow \mathbf{SLR}_{i,m}(\mathbf{S}, E_{rr}^{SLE_i}, is\_moving^{na})$ 
20:     $\forall n, \forall k, B_k^n \leftarrow m_n A_k^n$ 
21:     $\delta s_i^{Pred} \leftarrow \sum_n \sum_k w_{kn}^{LM S^n} B_k^n$ 
22:     $\Delta s_i^{Pred} \leftarrow \Delta s_i^{Pred} + \delta s_i^{Pred}$ 
23:    return  $\{\mathbf{S}^{Des}, move_m\}$ 
24:  end loop

```

Afin donc de pouvoir évaluer les performances du principe que nous avons décrit, et donc d'un recrutement basé sur la variation de configuration sensorielle, nous avons choisi d'implémenter un algorithme du SLR permettant de sélectionner les variations de senseurs nécessaires à l'affinage de la loi d'activation des catégories. Le principe de l'algorithme repose sur l'idée que puisqu'une variation de configuration sensorielle est une conjonction de facteurs nécessaires, à l'image de la loi d'activation de la catégorie qui en est issue, il suffit, pour tester la nécessité d'une condition (c'est-à-dire d'une variation d'un senseur particulier), de ne pas satisfaire une seule des conditions et de vérifier si cela met bien en échec l'ensemble de la prédiction associée à cette catégorie. L'activation de la catégorie c étant de la forme $A^c = A_i^c \times \dots \times A_k^c$,

alors $\forall j | A_j^c = 0, A^c = 0$. L'idée est alors de placer un unique senseur dans une position où sa contribution à l'activation sera nulle, en l'occurrence en s_j^0 , d'effectuer le mouvement associé (puisque chaque SLR n'est associé qu'à un unique couple senseur prédit - moteur, voir la figure 4.2), puis, en surveillant que l'activation reste nulle durant le mouvement, d'observer si celui-ci provoque un signal d'erreur E_{rr} (ligne 5 de l'Algorithme 3). L'activation étant nulle, s'il y a une erreur, il sera alors considéré que l'activation de cette catégorie était bien nécessaire pour que la prédiction soit bonne, et donc qu'il est nécessaire de prendre en compte ce senseur en entrée de cette catégorie. S'il n'y a pas d'erreur et que donc la prédiction est correcte, on retire la contribution de ce senseur à la loi d'activation de la catégorie. De plus, à chaque recrutement et avant ces tests sur les entrées nécessaires, la capacité de la nouvelle catégorie à permettre une prédiction correcte à partir de la configuration sensorielle courante, où donc l'erreur a été détectée, est elle-même testée, puisque c'est dans cette configuration que la nouvelle catégorie devra être pleinement activée. Pour cela, une éventuelle erreur de prédiction est là aussi surveillée, en poursuivant le mouvement ayant provoqué l'erreur tout en utilisant le correctif apporté par la nouvelle catégorie.

En résumé, le pseudo-code est donc le suivant, suite au recrutement d'une nouvelle catégorie :

- 1 Bouger (pour observer si l'erreur se reproduit)
 - Si l'erreur est reproduite : on conserve la catégorie
 - Si l'erreur n'est pas reproduite : on efface la catégorie et on quitte l'algorithme de recrutement
- 2 *Pour toutes les conditions nécessaires :*
 Mettre seulement cette condition à s^0 (la catégorie est ainsi désactivée) et les autres à s^1
ET
 Bouger
 - L'erreur n'est plus présente : on garde cette condition
 - L'erreur est toujours présente : on retire cette condition

L'Algorithme 4 montre en détails l'algorithme proposé de $SLR_{i,m}$, pour le senseur i et le moteur m . Notons que cet algorithme, destiné à tester et éprouver ce mode de recrutement, effectue systématiquement, et de manière ordonnée, la séquence d'actions nécessaire à la validation ou l'invalidation de chaque lien en entrée de la catégorie. Il faut cependant considérer que cette manière de procéder est à penser en parallèle avec d'autres systèmes de sélection de l'action, et entre en compétition avec d'autres types de comportements (de type réflexe pour la survie, par exemple). Ainsi la capacité qu'un robot aura à avoir un comportement proche de celui décrit dans cet algorithme devra faire l'objet d'un développement progressif de compétences diverses (comme, par exemple, une capacité d'inhibition liée au développement de buts extrinsèques et intrinsèques). Notons également que l'utilisation, ou pas, de certains liens en entrée n'a pas à être binaire, mais pourra par exemple être fonction d'une mesure de confiance plus élaborée que celle décrite ici. Ces questions seront de nouveau discutées en conclusion (sec. 4.4), et nous présentons ici un algorithme uniquement dédié au fonctionnement d'un tel mécanisme de recrutement.

Algorithm 4 : SLR

```

1:  $C^{tested} = \{\emptyset\}$ 
2:  $Step = 0$ 

3: function SLRi,m( $\mathbf{S}, E_{rr}^{SLE_i}, is\_moving$ )
4:   CHECK_NEW_CATEGORY( $\mathbf{S}, E_{rr}^{SLE_i}, is\_moving$ )
5:    $\{move_m, \mathbf{S}^{Des}\} \leftarrow \text{PLAY}()$ 
6:   // Compute vector of activations  $\mathbf{A}$ , (see eq.4.1)
7:    $\forall C, \forall i \in Sen(C), A_i^c \leftarrow \prod_i A_i^c$ , with  $A_i^c = \mathcal{R} \left[ \frac{s_i - s_i^0}{s_i^1 - s_i^0} \right]$ 
8:   return  $\{\mathbf{A}, \mathbf{S}^{Des}, move_m\}$ 
9: end function

10: function CHECK_NEW_CATEGORY( $\mathbf{S}, E_{rr}^{SLE_i}, is\_moving$ )
11:   // Initialise a new category when an error is detected
12:   if  $E_{rr}^{SLE_i}() \neq 0$  and  $is\_moving$  and  $C^{tested} = \{\emptyset\}$  then
13:     if  $E_{rr}^{SLE_i}() = 1$  then
14:       // An error is detected in the SLE, while activating motor  $m$ . A new category is recruited.
15:       Category  $C$  is created ;  $Sen(C) = \{\emptyset\}$ 
16:       for all  $s_i$  in  $\mathbf{S}$  do
17:         if  $|s_i - s_i^{stored}| > Th_1^{Err}$  then
18:           Add  $i$  to  $Sen(C)$ 
19:            $s_i^0 \leftarrow s_i^{stored}$  ;  $s_i^1 \leftarrow s_i$ 
20:            $Conf^{Link}(A_i^c) \leftarrow 0$ 
21:         end if
22:       end for
23:        $Conf^{Cat}(C) \leftarrow 0$ 
24:     else if  $E_{rr}^{SLE_i}() = -1$  then
25:        $\mathbf{S}^{stored} \leftarrow \mathbf{S}$ 
26:     end if
27:   end if
28: end function

29: function PLAY()
30:    $move_m \leftarrow 0$  ;  $\mathbf{S}^{Des} \leftarrow \{\emptyset\}$ 
31:   if  $C^{tested} = \{\emptyset\}$  then
32:      $Step = 1$ 
33:      $C^{tested} \leftarrow \text{SELECT\_CATEGO\_TO\_TEST}(C^{min})$ 
34:     if  $C^{tested} = \{\emptyset\}$  then
35:       return
36:     end if
37:   end if
38:   if  $Step = 1$  then
39:      $\mathbf{S}^{Des} \leftarrow \text{STEP\_1}()$ 
40:   else if  $Step = 2$  then
41:      $move_m \leftarrow \text{STEP\_2}()$ 

```

```

42:   else if  $Step = 3$  then
43:      $\mathbf{S}^{Des} \leftarrow STEP\_3()$ 
44:   else if  $Step = 4$  then
45:      $move_m \leftarrow STEP\_4()$ 
46:   end if
47:   return  $\{move_m, \mathbf{S}^{Des}\}$ 
48: end function

49: function SELECT_CATEGO_TO_TEST()
50:   if  $\prod_i Conf^{Link}(A_i^c) < Th^{conf}$  then
51:     return  $\arg \min_i Conf^{Link}(A_i^c)$ 
52:   end if
53:   return  $\{\emptyset\}$ 
54: end function

55: function STEP_1()
56:   //STEP 1 : Go in a configuration where the category is activated (where  $A^c = 1$ )
57:   for all  $i \in Sen(C^{min})$  do
58:      $s_i^{Des} \leftarrow s_i^1$  ; Add  $s_i^{Des}$  to  $\mathbf{S}^{Des}$ 
59:   end for
60:   if  $\mathbf{S}^{Des}$  is achieved or timeout then
61:     if  $Conf^{Cat}(C) = 0$  then
62:        $Step = 2$ 
63:     else
64:        $Step = 3$ 
65:     end if
66:     return  $\{\emptyset\}$ 
67:   else
68:     return  $\mathbf{S}^{Des}$ 
69:   end if
70: end function

71: function STEP_2()
72:   //STEP 2 : Move to test whether the correction won't create a mistake
73:   if  $E_{rr}^{SLE_i}() = 1$  or timeout then
74:     // Prediction error, the robot cannot reproduce the error. The category is erased.
75:     Erase category  $C$ 
76:     return 0
77:   else if  $E_{rr}^{SLE_i}() = -1$  then
78:     // Prediction is good : this new correction is necessary
79:      $Conf^{Cat}(C) \leftarrow 1$ 
80:      $Step = 3$ 
81:     return 0
82:   end if
83:   return 1
84: end function

```

```

85: function STEP_3()
86:   // STEP 3 : Select a condition and put the related sensor at  $s^0$  to test it
87:   Select one sensor  $i$  in  $Sen(C^{min})$ 
88:   Add  $s_i^0$  to  $S^{Des}$ 
89:   if  $S^{Des}$  is achieved or timeout then
90:     Step = 4
91:     return  $\{\emptyset\}$ 
92:   end if
93:   return  $S^{Des}$ 
94: end function

95: function STEP_4()
96:   // STEP 4 : move when just one condition is not satisfied ( $A_i^c = 0$ )
97:   if  $E_{rr}^{SLE_i}() = 1$  or timeout then
98:     // STEP 4b : Prediction error, then the category should have been activated : this link is not
     necessary for its activation
99:     Remove  $i$  to  $Sen(C^{min})$ 
100:    Step = 1
101:    return 0
102:   else if  $E_{rr}^{SLE_i}() = -1$  then
103:     // STEP 4a : Good prediction : this link is a necessary condition for the associated correc-
     tion
104:      $Conf^{Link}(A_i^c) \leftarrow 1$ 
105:     Step = 1
106:     return 0
107:   end if
108:   return 1
109: end function

```

Le $SLR_{i,m}$ prend en entrée le vecteur de tous les senseurs \mathbf{S} (tout comme le SLE auquel il est attaché), l'erreur de prédiction $E_{rr}^{SLE_i}$ constatée par le SLE_i , ainsi que la variable binaire is_moving , indiquant si seul ce moteur m était activé lorsque l'erreur a été constatée (indication nécessaire pour pouvoir attribuer l'erreur au bon couple sensorimoteur). Supposons ici que l'erreur de prédiction puisse prendre 3 valeurs : 0 si l'erreur n'est pas vérifiée (algorithme 3 ligne 5), 1 si il y a une erreur de prédiction, -1 si la prédiction est bonne. En sortie du $SLR_{i,m}$ on trouve le vecteur correspondant à l'activation de chaque catégorie A , à l'image de ce que l'on trouve dans le SAW ou le WTA. Mais il y a en plus un vecteur de configuration de senseurs désirés \mathbf{S}^{Des} , ainsi qu'une variable $move_m$ permettant l'activation du moteur m associé. Ces sorties supplémentaires sont utilisées pour tester les entrées pertinentes d'une catégorie (rappelons ici que ces sorties sont semblables à celles du SLS et qu'elles seront utilisées de la même manière).

Brièvement, l'Algorithme 4 du SLR commence par initialiser une éventuelle nouvelle catégorie (fonction CHECK_NEW_CATEGORY). Puis, le mécanisme de jeu (de la fonction PLAY) détermine si une catégorie requiert un affinage de son apprentissage (fonction SELECT_CATEGO_TO_TEST). Si c'est le cas, les quatre étapes du jeu pourront alors alterner (fonctions STEP_1, STEP_2, STEP_3, STEP_4). Enfin, le calcul de l'activation de chaque catégorie est effectué (ligne 7).

Examinons maintenant l'Algorithme 4 plus en détail. Le SLR effectue 3 fonctions différentes : d'une part, il crée et initialise de nouvelles catégories (fonction CHECK_NEW_CATEGORY). D'autre part, il gère les demandes de configurations sensorielles et d'actions qui permettront de préciser les conditions d'activation de ces nouvelles catégories (fonction PLAY). Enfin, il détermine l'activation courante de chacune des catégories (ligne 7). Le SLR commence donc par observer l'erreur de prédiction, $E_{rr}^{SLE_i}$. Si une erreur est constatée pendant que le moteur associé était en mouvement, alors une catégorie C est créée, avec une loi d'activation qui est fonction de tous les senseurs ayant suffisamment varié (comme dans l'exemple montré dans la figure 4.1), et ces senseurs sont stockés dans $Sen(C)$. Afin de déterminer ceux-ci, nous comparons pour chaque senseur la distance entre sa valeur stockée et sa valeur courante : si cette distance dépasse le seuil Th_1^{Err} alors ce senseur est ajouté à la liste des senseurs $Sen(C)$ dont dépend l'activation de C , c'est-à-dire à la liste des senseurs qui participent à la loi d'activation de la nouvelle catégorie. Notons que nous utilisons ici le même seuil que celui utilisé par le SLE pour détecter une erreur (voir ligne 6 de l'Algorithme 3). À la ligne 19, les valeurs utilisées par la règle d'activation (eq.4.1) sont initialisées comme suit : les s^1 prennent les valeurs courantes, et les s^0 prennent les valeurs stockées lors de la précédente bonne prédiction du SLE (lorsque $E_{rr}^{SLE_i} = 1$), voir ligne 25. De plus, on associe à chaque règle d'activation partielle du senseur A_i^c une confiance notée $Conf^{Link}(A_i^c)$, initialisée à 0, reflétant la confiance en l'existence d'un lien entre la catégorie C venant corriger l'erreur, et une entrée sensorielle donnée i . Plus la participation de ce senseur sera expérimentalement identifiée comme étant une condition nécessaire, plus cette confiance augmentera (mais notons que sa valeur est binaire dans notre algorithme). Finalement, la catégorie elle-même est dotée d'une confiance $Conf^{Cat}(C)$ permettant de valider ou d'invalider l'ensemble de la catégorie C elle-même, et de la correction de prédiction qui y est associée. Elle reflète la confiance en l'existence de la catégorie elle-même, ce qui est testé par le tout premier mouvement, lorsque l'erreur vient d'être constatée.

Le SLR teste ensuite sa confiance générale, ligne 50, c'est-à-dire le produit des confiances en l'existence de chaque lien, entre chaque catégorie et chaque senseur en entrée (dont la liste est stockée dans les $Sen(C)$ de chaque C). Selon la confiance désirée (déterminée par un seuil Th^{conf} , que nous avons fixé à 1 dans nos expériences), chaque condition, ou règle d'activa-

tion liée à un senseur donné (une causalité) sera testée de la manière exposée ci-après. Pour la catégorie comportant la confiance $Conf(A_i^c)$ la plus faible (ligne 51), on procède aux étapes suivantes :

- 1 On met tous les senseurs aux positions s^1 (ligne 57- 58). Un fois qu'ils y sont, alors si la confiance en la catégorie $Conf^{Cat}(C)$ n'a pas été testée on va à l'étape 2, sinon on va à l'étape 3.
- 2 Un mouvement est demandé afin de tester l'efficacité de la correction (ligne 83) dans la configuration sensorielle courante. S'il y a une erreur de prédiction, la catégorie est supprimée et l'algorithme du SLR s'arrête là (ligne 75), sinon on va à l'étape 3.
- 3 Un par un, chaque senseur i de $Sen(C)$ est successivement sélectionné pour être testé dans les étapes qui suivent. Pour cela, le senseur sélectionné est mis en position s_i^0 (ligne 88). De ce fait, la condition attachée à ce senseur n'est pas validée : la loi d'activation partielle devient nulle, et de fait l'activation de la catégorie devient nulle aussi. On va ensuite à l'étape 4 pour tester la nécessité de ce lien.
- 4 L'activation du moteur ayant provoqué l'erreur est demandée (ligne 108).
 - a S'il n'y a pas d'erreur à l'issue de ce mouvement : cela signifie que la condition sur le senseur i est bien nécessaire (ligne 104). Ce dernier reste dans la liste des senseurs qui conditionnent l'activation de la catégorie de C $Sen(C)$.
 - b S'il y a une erreur à l'issue de ce mouvement : cela signifie que la condition sur le senseur i n'est pas nécessaire, car, même avec ce senseur en position s_i^0 , la catégorie aurait du être activée. Il est donc retiré de $Sen(C)$ (ligne 99).

Dans tous les cas, la confiance associée à la nécessité de la participation de ce senseur à la catégorie augmente (elle passe à $Conf(A_i^c) = 1$). On retourne alors à l'étape 1.

Enfin, le SLR détermine l'activation courante de chaque catégorie C , en effectuant le produit des activations partielles A_i^c relatives aux senseurs de $Sen(C)$ (voir l'eq.4.1 et ligne 7).

4.3 Expériences

Dans cette partie, les expériences sont réalisées en simulation, afin de tester en parallèle, et sur de longues périodes, différents modèles ayant les mêmes entrées (et donc au même titre que dans la section 3.1.4). Il s'agira de simuler un bras robotique, et d'apprendre la relation entre la proprioception et la position visuelle de l'extrémité du bras robotique. Pour cela, nous avons modélisé la relation entre moteurs, proprioception et position du bras, tel que schématisé dans la figure 4.4. Puisque seule la relation entre proprioception et position visuelle du bras nous intéressera dans ces simulations, notons que nous avons supposé une relation linéaire entre l'activation motrice et la proprioception. Afin d'être plus réaliste, cette relation devrait faire l'objet d'un apprentissage par des SLEs (ou plus simplement être contrôlée par un PID), lesquels auront notamment appris les butées articulaires. Nous supposons donc ici que cette relation est connue, et que toute position désirée sur la proprioception pourra être obtenue par l'activation correspondante des moteurs. Dans ces expériences, les valeurs de la proprioception sont bornées entre 0 et 1.

La proprioception et la position du bras sont notées comme des senseurs, s , afin de marquer la différence fondamentale entre moteurs (à qui l'on assigne une magnitude) et senseurs (dont on reçoit une valeur). Cela permet également de marquer le fait qu'en tant que senseurs ils pourront faire l'objet de positions désirées, ou servir à catégoriser d'autres SLEs, et, pour finir, de souligner le fait que s'ils sont suffisamment bien prédits par un SLE, leurs variations pourront servir d'entrées motrices à d'autres SLEs. Cette dernière propriété sera d'ailleurs sollicitée, dans la mesure où les variations de la proprioception seront de fait considérées comme étant autant de moteurs. Cette situation est équivalente à celle où d'autres SLEs auraient prédit, avec suffisamment de certitude, les variations de la proprioception, et que les variations de celle-ci pouvaient désormais être considérées comme autant de moteurs.

La relation entre la proprioception et la position du bras fait, elle, l'objet d'une loi qui pourra être non linéaire, et que nous ferons évoluer au cours de nos simulations pour observer son apprentissage par des SLEs.

Le schéma de la figure.4.4 modélise ces relations. Au cours des expériences, nous précisons chaque fois quels senseurs et moteurs nous utiliserons en utilisant ces notations, ainsi que la loi implémentée.

Notons, pour finir, que nous avons choisi d'avoir deux SLRs pour chaque "direction", positive ou négative, de moteurs, afin de prendre en compte le fait que la loi ne soit pas nécessairement la même lorsque le bras bouge dans un sens ou dans un autre. Ce choix peut être justifié, par exemple, par la présence de butée articulaire, ou encore par les effets de la pesanteur, qui empêche de présupposer une équivalence des lois liées aux magnitudes positive et négative.

Dans un premier temps (Sec. 4.3.1), nous effectuerons une série d'expériences visant à détailler le comportement du modèle lors de cas simples, puis, dans un second temps (Sec. 4.3.2), nous observerons ses performances à l'aide d'une simulation du bras Katana, en les comparant à celles du modèle absolu.

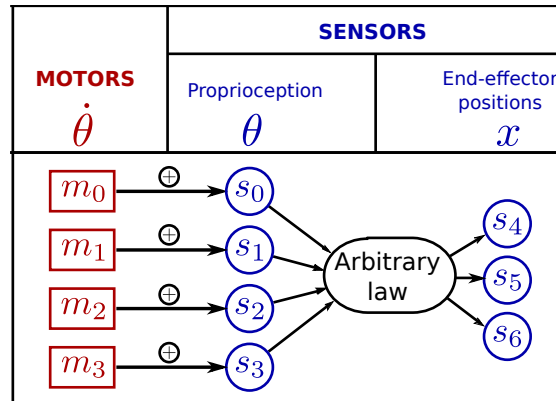


FIGURE 4.4 – Représentation schématique des relations entre les moteurs, la proprioception et la position de l’extrémité du bras. Les notations seront reprises dans les expériences, et une loi sera chaque fois définie.

4.3.1 Mécanisme de recrutement

Simulation 1 : processus de recrutement lors d’une non-linéarité

Afin d’observer expérimentalement les étapes du processus d’apprentissage des lois, nous avons commencé par travailler avec une simulation de bras robotique simple, à une seul degré de liberté, où la relation entre la position de l’extrémité x du bras et l’articulation θ (variant entre 0 et 1) suit la même loi non linéaire que celle décrite dans la section 3.1.3, à savoir :

$$x = \begin{cases} 0.5 \cdot \theta & \text{si } \theta < 0.5 \\ 1.5 \cdot \theta - 0.5 & \text{sinon.} \end{cases} \quad (4.4)$$

Dans cette première expérience, nous n’utiliserons qu’un seul moteur sur les 4 : m_0 . Comme nous l’avons précédemment expliqué, ce moteur ne fait pas varier directement le senseur mais la proprioception associée, selon une loi supposée ici linéaire (un autre SLE devrait théoriquement être dédié à cet apprentissage). En l’occurrence, la proprioception liée au moteur m_0 est le senseur s_0 , et correspondra au θ dans l’équation 4.10. Ainsi l’activation du moteur m_0 accroît la valeur de la proprioception correspondante s_0 . Notons encore que dans cette expérience, la valeur de la proprioception est bornée entre 0 et 1. Le senseur correspondant à la position de l’extrémité du bras, et dont le SLE va prédire les variations, est le senseur s_4 et correspond à x dans l’équation 4.10. Pour résumer, nous avons donc $m_0 = \dot{\theta}$, $s_0 = \theta$ et $\delta s_0 = \dot{x}$.

Notons également que dans ces expériences, les prescriptions sur la proprioception sont linéairement traduites en activité motrice (laquelle est bornée à 0.01), et que toute activité motrice provenant d’une prescription du SLR inhibera automatiquement le babillage moteur.

Dans cette expérience, il a été empiriquement choisi les paramètres suivants : $Th_1^{Step} = 0.1$ (et $Th_1^{Err} = Th_1^{Step}/2 = 0.05$). Nous allons dérouler ici, et à l’aide de la figure 4.5, les différentes étapes du recrutement telles que décrites dans l’algorithme, pour le SLE_{s_4} prédisant les variations de s_4 :

- Le moteur commence son babillage (voir fig.4.5 courbe c) en bleu et plein). L’activité motrice positive entraîne immédiatement une augmentation linéaire de la valeur de la

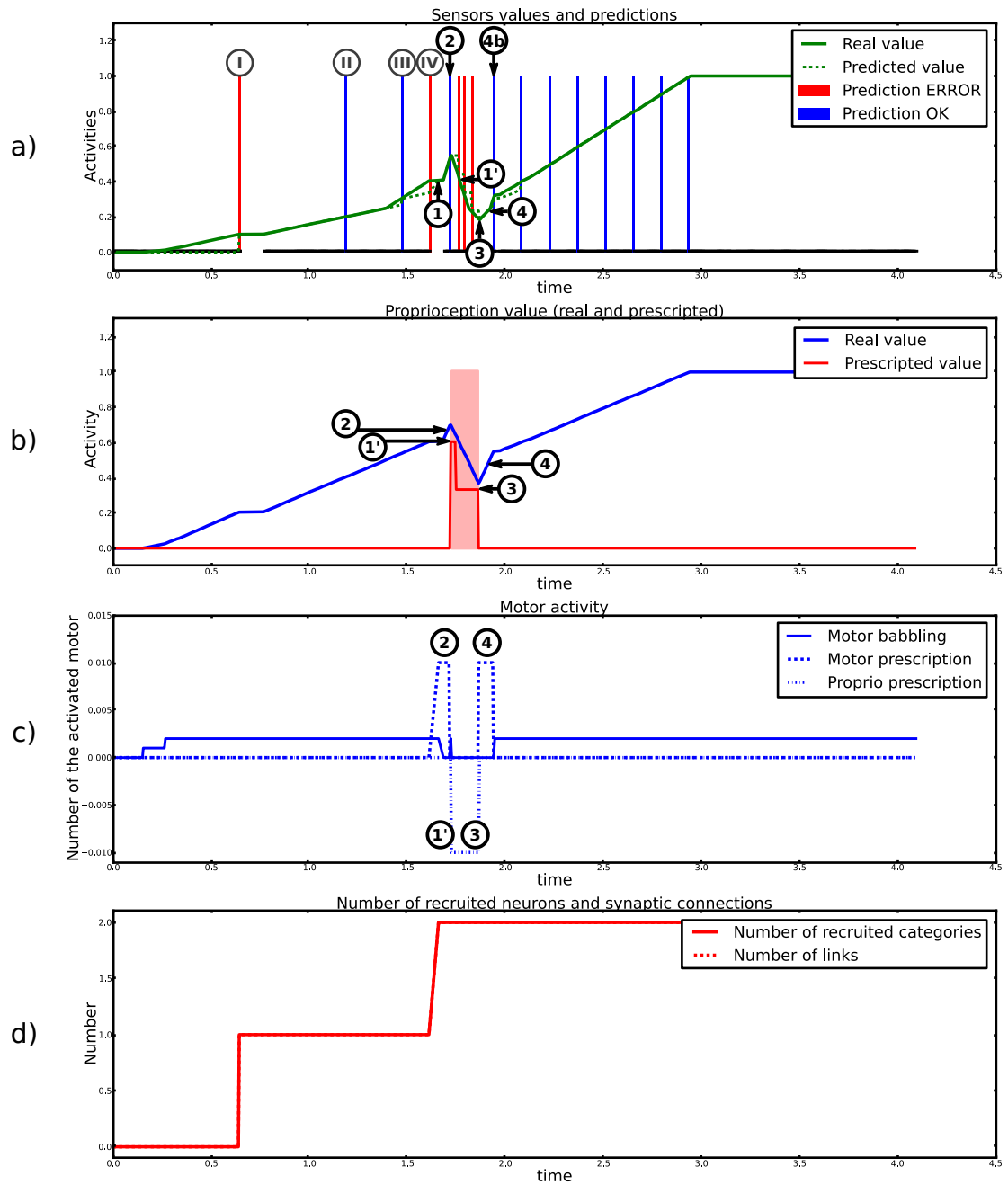


FIGURE 4.5 – Recrutements du SLR avec la loi suivant l’eq.4.10. La courbe a) représente en vert la valeur réelle du capteur s_4 , en pointillé la valeur prédite, et les barres bleues et rouges indiquent respectivement les prédictions correctes ou les erreurs de prédiction. La courbe b) montre en bleu la valeur de la proprioception s_0 , la zone rouge indique les moments où ce capteur reçoit une valeur désirée, et la courbe rouge indique cette valeur. La courbe c) indique l’activité motrice provenant de babillage moteur en bleu plein, d’une prescription motrice en pointillé, et d’une prescription sur la proprioception en trait et pointillé. La courbe d) indique en trait plein le nombre de catégories recrutées par ce SLE, et en pointillé le nombre de liens total.

proprioception s_0 (voir courbe b) en bleu). Dans le même temps, le senseur (s_4) augmente suivant l'eq.4.10 (voir courbe a) en vert).

- Une erreur est détectée en a)-**I** (la première barre rouge).
 - En effet, la valeur réelle de s_4 est de 0.1005 et celle prédite est de 0 (or $|0.1005 - 0| > Th_1^{Err}$).
 - Une nouvelle catégorie est recrutée (voir la courbe d). Comme il s'agit du premier mouvement effectué, il s'agit du cas particulier évoqué précédemment (sec. 4.2.1) : la loi d'activation est donc $A_c = 1$. Et la valeur du lien conditionnel du LMS est de 0.5025.
 - Puisqu'il n'y a aucun lien à tester dans ce cas particulier, la phase de recrutement est immédiatement terminée. La figure 4.6 montre une illustration du réseau à la fin de ce premier recrutement. Notons que dans le décompte du nombre total de liens recrutés, ce type de catégorie compte tout de même pour un lien (voir la courbe d).
- Une prédiction correcte est détectée en a)-**II**. En effet, la valeur réelle est de 0.2005 et celle prédite est de 0.2. Les valeurs courantes de **S** sont enregistrées dans S^{Stored} .
- Une autre prédiction correcte est détectée en a)-**III**. Cette fois pourtant, la loi a changé depuis la précédente vérification (d'après l'eq.4.10, quand $\theta = s_0 = 0.5$ soit quand $x = s_4 = 0.25$), mais pas suffisamment tôt pour provoquer une erreur : la valeur réelle est de 0.302500 et celle prédite est de 0,267835, et $|0.302500 - 0,267835| < Th_1^{Err}$. De la même manière, les valeurs courantes de **S** sont enregistrées dans S^{Stored} (et notamment, $s_0^{Stored} = 0.537$).
- Une seconde erreur est détectée en a)-**IV**.
 - En effet la valeur réelle est 0.404499 et celle prédite est 0,33667 et $|0.404499 - 0,33667| > Th_1^{Err}$.
 - Une nouvelle catégorie est recrutée (voir la courbe d), et la valeur du lien conditionnel du LMS est de 1.027727.
 - Seul le senseur s_0 a varié. En a)-**III**, lors de la précédente prédiction correcte, sa valeur était de $s_0^{Stored} = 0.537$ et elle est ici de $s_0 = 0.604999$.
 - Ainsi la loi d'activation de cette catégorie est $A^c = A_{s_0}^c$, définit selon l'équation 4.1 avec $s_{s_0}^0 = 0.537$ et $s_{s_0}^1 = 0.604999$ (voir la figure 4.7 illustrant le réseau à la fin du second recrutement).
 - Les étapes précédemment décrites de l'algorithme s'ensuivent :
 - * **Étape 1** : Le SLR cherche à aller à la position où la catégorie est activée. Or nous avons déjà $s_0 = 0.604999 = s_{s_0}^1$, il n'y a donc rien à faire à cette étape.
 - * **Étape 2** : Le moteur est activé afin de tester si la correction apprise est bien requise, et une prédiction correcte est détectée, voir courbe c)(2). La catégorie est donc gardée et $Con.f^{Cat}(A^c) = 1$.
 - * **Étape 3** : le SLR répète **Étape 1** (1') et atteint $s_{s_0}^1 = 0.604999$, voir courbe c)(1'). Puis il désire mettre l'une des valeurs à s^0 . Comme il n'y en a qu'une

ici, il désire donc mettre s_0 à la valeur $s_{s_0}^0 = 0.537$, ce que l'on observe sur la courbe b)(3). Le sensor atteint cette valeur par une activité motrice correspondante dans m_0 , voir courbe c)(3).

- * **Étape 4** : Le moteur est activé, voir courbe c)(4).
- * **Étape 4a** : Il n'y a pas d'erreur, donc la variation du sensor s_0 est considérée comme une cause nécessaire au changement de contexte. Sa confiance passe donc à $Conf^{Link}(A^C) = 1$, et il reste dans la liste des sensors qui sont des conditions.

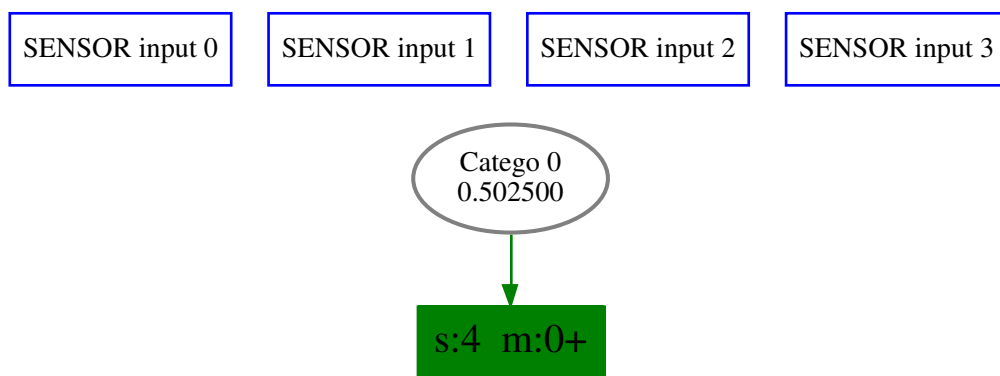


FIGURE 4.6 – Schéma représentant les catégories du SLR du moteur m_0 et du sensor s_4 , pour les mouvements positifs. La valeur associée à la catégorie (0.5025) indique le poids appris par le LMS (voir fig.4.2). Il s'agit ici du recrutement de la première catégorie (et donc du cas particulier, avec une activation constante).

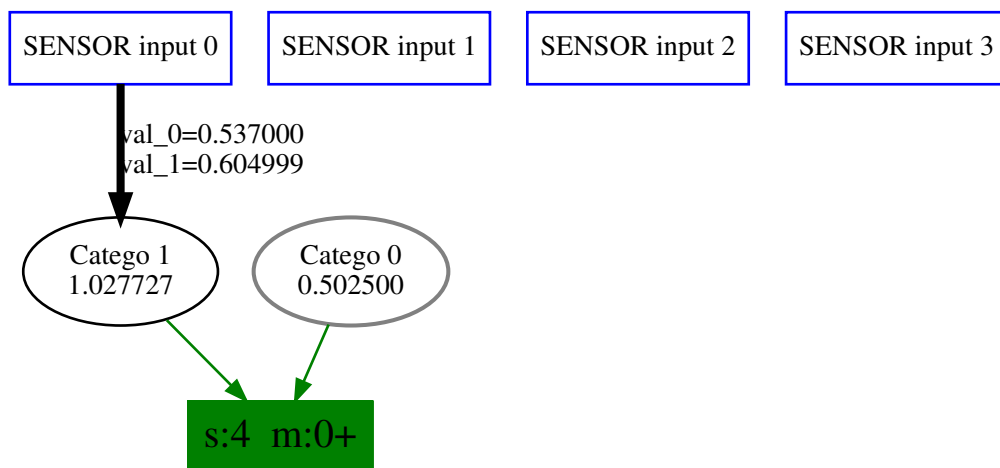


FIGURE 4.7 – Schéma représentant les catégories du SLR du moteur m_0 et du sensor s_4 , pour les mouvements positifs. La valeur associée à la catégorie (1.027727) indique le poids appris par le LMS (voir fig.4.2). Le lien allant de “SENSOR input 0” à “Catego 1” indique que la catégorie recrutée possède un lien vers le sensor s_0 , avec $s_{s_0}^0 = val_0$ et $s_{s_0}^1 = val_1$. Il s'agit ici du recrutement de la seconde catégorie.

La figure 4.7 nous montre l'état du réseau au terme de cet apprentissage. Puisque nous avons $m_0 = \dot{\theta}$, $s_0 = \theta$ et $\delta s_4 = \dot{x}$, ce réseau peut se traduire par l'équation suivante :

$$\dot{x} = 0.5025 \cdot \dot{\theta} + 1.027727 \cdot \mathcal{R} \left[\frac{\theta - 0.5025}{0.604999 - 0.5025} \right] \cdot \dot{\theta} \quad (4.5)$$

Nous voyons donc que ce réseau constitue une approximation de l'équation 4.10, dont la précision dépend du seuil Th_1^{Err} qui aura été ici arbitrairement fixé à 0.1.

Notons ici une des particularités de ce mode de recrutement. Nous avons précédemment parlé de l'“oubli catastrophique” en indiquant que ce modèle était notamment fait pour tenter d'apporter une solution à cet écueil. Si les précédents apprentissages sont ici conservés, notons que par définition, chaque recrutement partitionnant l'espace des senseurs en deux, crée une frontière entre une zone d'inactivation et une zone d'activation. Par définition donc, il y a un impact global sur la moitié de cet espace. Bien que le mode de recrutement soit fait pour éviter qu'il y ait “oubli”, il peut y avoir des cas, dépendants notamment de l'exploration sensorimotrice effectuée, où des zones de l'espace avaient de bonnes prédictions et qui, suite à un recrutement, peuvent devenir mauvaises.

Ainsi le mode de recrutement crée des frontières dans l'espace des senseurs, mais par couple senseur-moteur, et par une observation ponctuelle de l'environnement. La comparaison de la prédiction avec la réalité, et le recrutement qui s'en suit, sont fonction de deux choses :

- Cette comparaison a lieu lorsqu'une variation du senseur (ou de la valeur prédite) est supérieure à un seuil donné : $|\Delta s_i^{Real}| > Th_1^{Step}$ **or** $|\Delta s_i^{Pred}| > Th_1^{Step}$ (ligne 5 de l'Algorithme 3).
- Le moteur associé (appelé m_{na} dans l'Algorithme 3) est le seul à être activé.

Or, un senseur peut varier indépendamment de ce moteur là (ce qui arrive d'autant plus souvent que nous avons choisi de créer deux moteurs correspondants aux activations positives et négatives du moteur, voir 3.1 et l'introduction de cette section 4.3). De ce fait, du point de vue de l'observation de ce couple senseur-moteur, et donc du suivi des prédictions correctes, c'est un peu comme si le senseur s'était “téléporté” ailleurs, dans la mesure où le moteur peut ne pas être à l'origine de sa variation. Ceci peut poser problème, car, comme nous l'avions évoqué, le recrutement définit des frontières entre des lois, qui seront d'autant plus efficaces lorsque l'exploration est continue dans cet espace, et que donc, lors d'un recrutement, la zone précise correspondant à la frontière réelle est apprise. Mais dans le cas de tels sauts dans l'espace, dus à l'activation d'autres moteurs, la frontière n'est que grossièrement définie, et aura de fait à être précisée par la suite.

Voyons, dans un nouvel exemple de la simulation qui vient d'être décrite, comment ce problème peut apparaître et se résoudre. Dans la figure 4.8, montrant le couple $s_4 - m_{0-}$, nous notons les étapes de recrutement suivantes :

- La catégorie 0 a été apprise lorsque $s_0 < 0.5$, et la dernière bonne prédiction était à $s_0 = 0.11$. Notons qu'avec cette seule catégorie, la loi est bonne pour toutes les activations de m_{0-} (donc négative) où $s_0 < 0.5$.
- Une activation maintenant positive du moteur fait varier sa valeur jusqu'à ce que $s_0 > 0.5$ (en le “téléportant” donc aux yeux du précédent couple senseur-moteur).

- Suite à cela, une nouvelle activation négative du moteur est effectuée, et de fait une nouvelle erreur est constatée, cette fois en $s_0 = 0.8$, et la catégorie 1 est alors recrutée. De fait, la frontière créée est une rampe entre $s_0^0 = 0.11$ et $s_0^1 = 0.8$, quand l'équation nous indique que la frontière réelle est précisément à 0.5. Nous voilà, en conséquence, dans le cas décrit précédemment, qui entraîne une sorte d'“oubli” de la loi réelle pourtant jusque là bien approximée, à mesure que cette catégorie s'active en s'approchant de $s_0 = 0.5$. Il y a donc ici une erreur qui peut apparaître. Mais cette erreur ne sera que locale, délimitée par 0.5 d'une part (car, au-delà, la loi change réellement), et d'autre part, par $s_0^0 = 0.11$. De fait, la manière dont se résout ce problème illustre la manière dont une loi locale est apprise : deux catégories doivent être recrutées, l'une pour délimiter où cette loi commence, et l'autre pour délimiter où elle ne s'applique plus, ou encore quand elle se modifie en une nouvelle loi.
- La vigilance étant relativement basse dans cet exemple (à 0.05), une nouvelle erreur apparaît effectivement, juste à la frontière (de 0.5 à 0.45), alors que la loi associée à la catégorie 1 est déjà partiellement activée. La catégorie 2 est donc recrutée pour corriger cette erreur, et délimite là où l'erreur locale commence. Mais cette catégorie, créée pour corriger l'activation de la catégorie 1 lorsque s_0 s'approche de 0.5, reste active pour toutes les valeurs inférieures à $s_0 = 0.45$, et donc quand la catégorie 1 est totalement désactivée.
- De fait, à mesure que le capteur se rapproche de 0, la contribution de la catégorie 2 est pleine, quand celle de la catégorie 1 devient nulle. Ainsi, la correction associée à la catégorie 2, créée pour corriger la loi associée à la catégorie 1, devient en trop. Une erreur apparaît, et la catégorie 3 est recrutée pour délimiter la fin de la correction de cette erreur locale (avec $s_0^0 = 0.21$ et $s_0^1 = 0.1$).

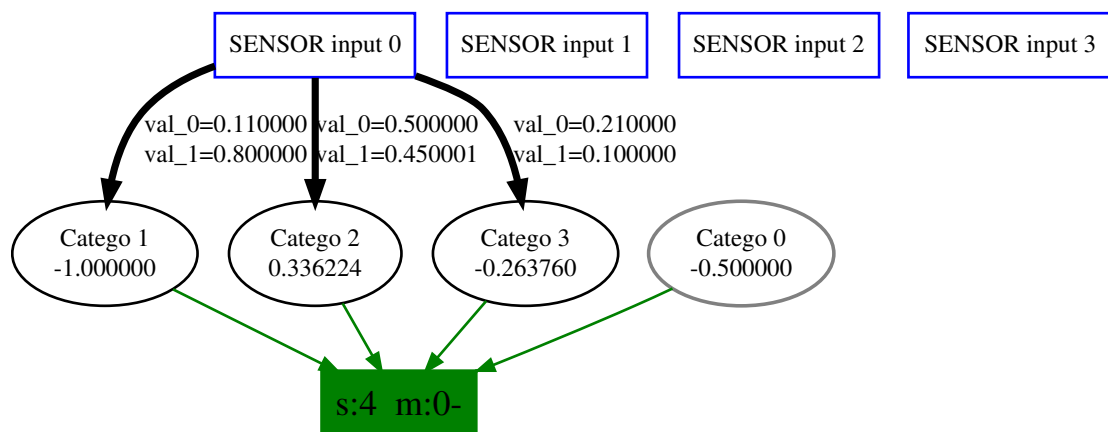


FIGURE 4.8 – Exemple de recrutement non optimal avec un “oubli” temporaire.

Ainsi, les prédictions peuvent être moins bonnes suite à un recrutement lorsqu'une frontière est trop peu précisément définie. De fait, il y a localement une mauvaise loi qui s'applique. Dans ce genre de cas, une correction uniquement locale doit être effectuée, ce qui se fait par le recrutement de 2 catégories et lois associées.

Notons que de mauvaises lois, en impactant les prédictions, influencent également les valeurs d'affordances (*i.e.* de simulation motrice). De ce fait le SLS, et le mécanisme de pondération par

l'efficacité (voir section 3.2.3) utilisés par le robot pour agir, peuvent potentiellement donner de moins bons résultats (lors d'une tâche d'atteinte par exemple). Ce sera le cas à plus forte raison lorsque la prédiction est de 0, ou même de signe opposé à la véritable loi, car alors le robot risque de ne pas bouger ou de bouger dans la mauvaise direction.

Simulation 2 : processus de recrutement lors de dépendances entre senseurs

Dans cette seconde expérience, la loi est la suivante :

$$\forall i, x^i = \theta_i \cdot (0.2 + 0.8 \cdot \theta') \quad (4.6)$$

$$\text{Avec } \theta' = \begin{cases} \theta_{i-1} & \text{si } i > 0 \\ 0 & \text{sinon.} \end{cases} \quad (4.7)$$

Cette loi permet de tester le cas où la loi prédisant les variations d'un senseur, dans notre cas s_5 , est apprise alors qu'un sous-ensemble d'autres senseurs varient mais que tous ne sont pas pertinents pour la prédiction (dans notre cas, seules les variations de s_1 seront pertinentes pour s_5). Pour la clarté des explications, la loi est ici linéaire. (Notons que, dans le dernier exemple, une simulation du bras Katana sera utilisée.) Cet exemple permettra d'illustrer le fait qu'un ensemble potentiellement grand de lois pourra être appris en ne contextualisant que le sous-ensemble des senseurs qui se trouvera avoir une réelle influence sur les variations du senseur prédit. Au cours de l'expérience nous ferons varier les senseurs permettant de faire varier la loi (s_0 dans notre cas), ainsi que d'autres senseurs ne faisant pas varier la loi. Par cet exemple simple nous cherchons à illustrer ce qu'il se passe par exemple lorsque la sensorimotricité du robot est apprise et qu'une perturbation, captable par un ou plusieurs senseurs (comme l'est la prise d'un outil), intervient et transforme l'ensemble des règles précédemment apprises selon une loi donnée. C'est donc avec un seul senseur que nous allons voir comment la transformation est apprise, par l'expérimentation. Afin de comprendre l'importance qu'il y a à tester les liens d'une nouvelle catégorie, nous allons également bouger un autre senseur n'ayant lui pas d'influence, en l'occurrence s_2 .

Rappelons que dans cette implémentation, durant une phase de recrutement et de tests d'un SLR, tout autre recrutement est inhibé.

L'expérience suivra donc les étapes suivantes, qui peuvent se voir sur la figure 4.9 :

- Les proprioceptions sont toutes à 0 au début de l'expérience.
- Le moteur m_1 est manuellement activé avec une valeur constante (trait plein dans la courbe d). La proprioception associée s_1 augmente linéairement, voir fig.4.9 b)-I. Suivant l'équation 4.9, le senseur s_5 augmente lui aussi linéairement, avec un facteur 0.2 (puisque la loi de s_5 est donc $s_5 = s_1 \cdot 0.2$ puisque selon l'équation 4.9 nous avons $s_5 = s_1 \cdot (0.2 + 0.8 \cdot s_0)$ et que $s_0 = 0$). La loi est apprise par le SLE_{s_5} , et la figure 4.11 montre une illustration du réseau à la fin de ce premier recrutement. Notons que suite à ce premier recrutement, une prédiction correcte a lieu, (voir la barre bleue en a)). Comme nous avons suivi les effets du SLR relatifs aux mouvements positifs, et que par la suite c'est le SLR associé au moteur négatif qui sera activé, ce sera donc la dernière activation correcte enregistrée par ce SLR : sont donc enregistrées, à cet instant, les valeurs $s_0^{Stored} = 0$, $s_1^{Stored} = 1$ et $s_2^{Stored} = 0$.

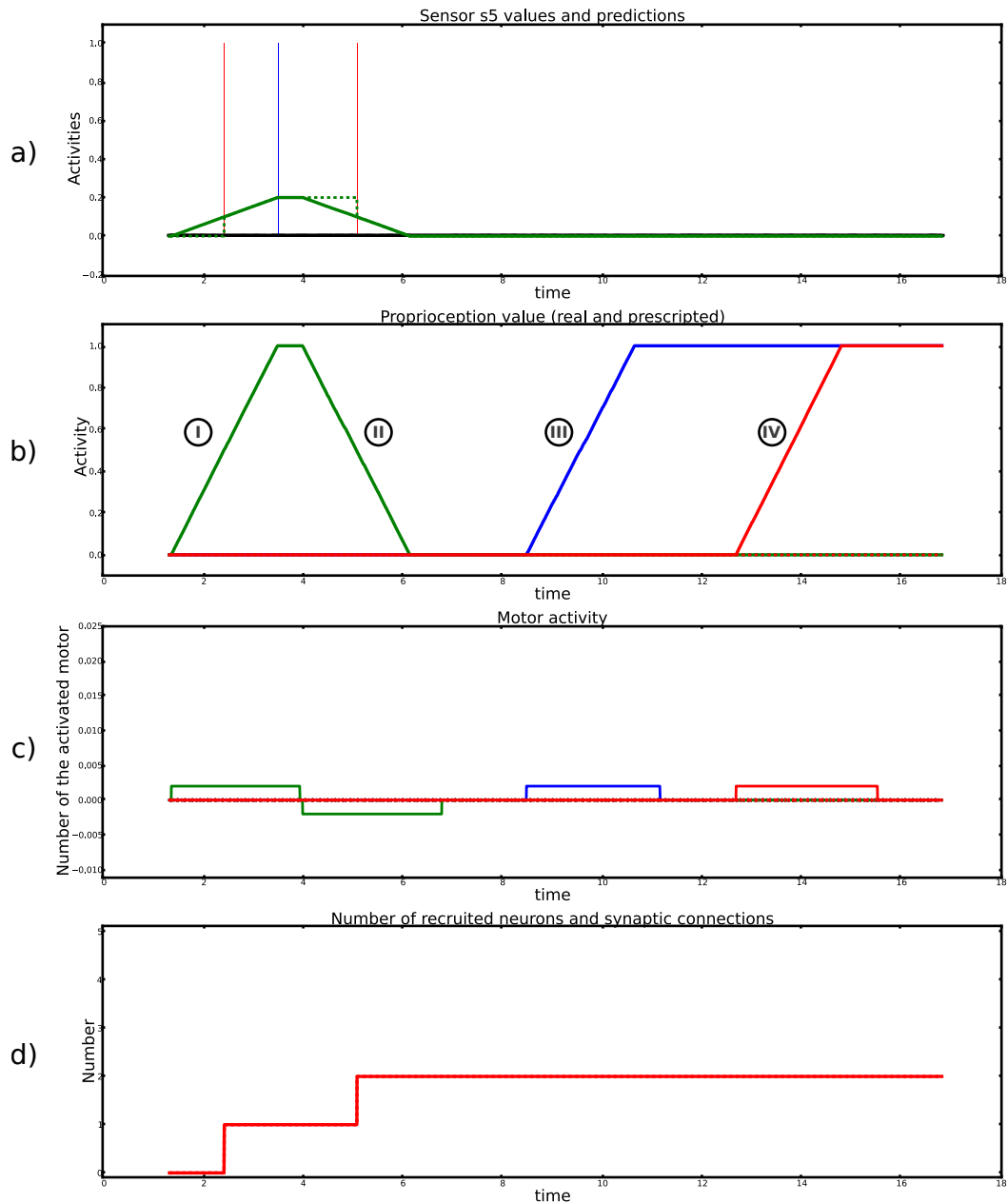


FIGURE 4.9 – Recrutements du SLR_{s_5} avec la loi suivant l'eq.4.9. La courbe a) représente en vert la valeur réelle du capteur s_4 , en pointillé la valeur prédite, et les barres bleues et rouges indiquent les prédictions correctes et les erreurs de prédiction, respectivement. La courbe b) montre les valeurs de la proprioception s_0 , s_1 et s_2 , respectivement en bleu, vert et rouge. Les zones colorées indiquent les moments où ces capteurs reçoivent une valeur désirée, et les courbes pointillées indiquent ces valeurs. La courbe c) indique les activités motrices provenant de m_0 , m_1 et m_2 respectivement en bleu, vert et rouge. Les activités issues d'un babillage moteur sont en trait plein, celles issues d'une prescription motrice en pointillé, et celles provenant d'une prescription sur la proprioception en tiret et pointillé. La courbe d) indique en trait plein le nombre de catégories recruté par ce SLE, et en pointillé le nombre de liens total.

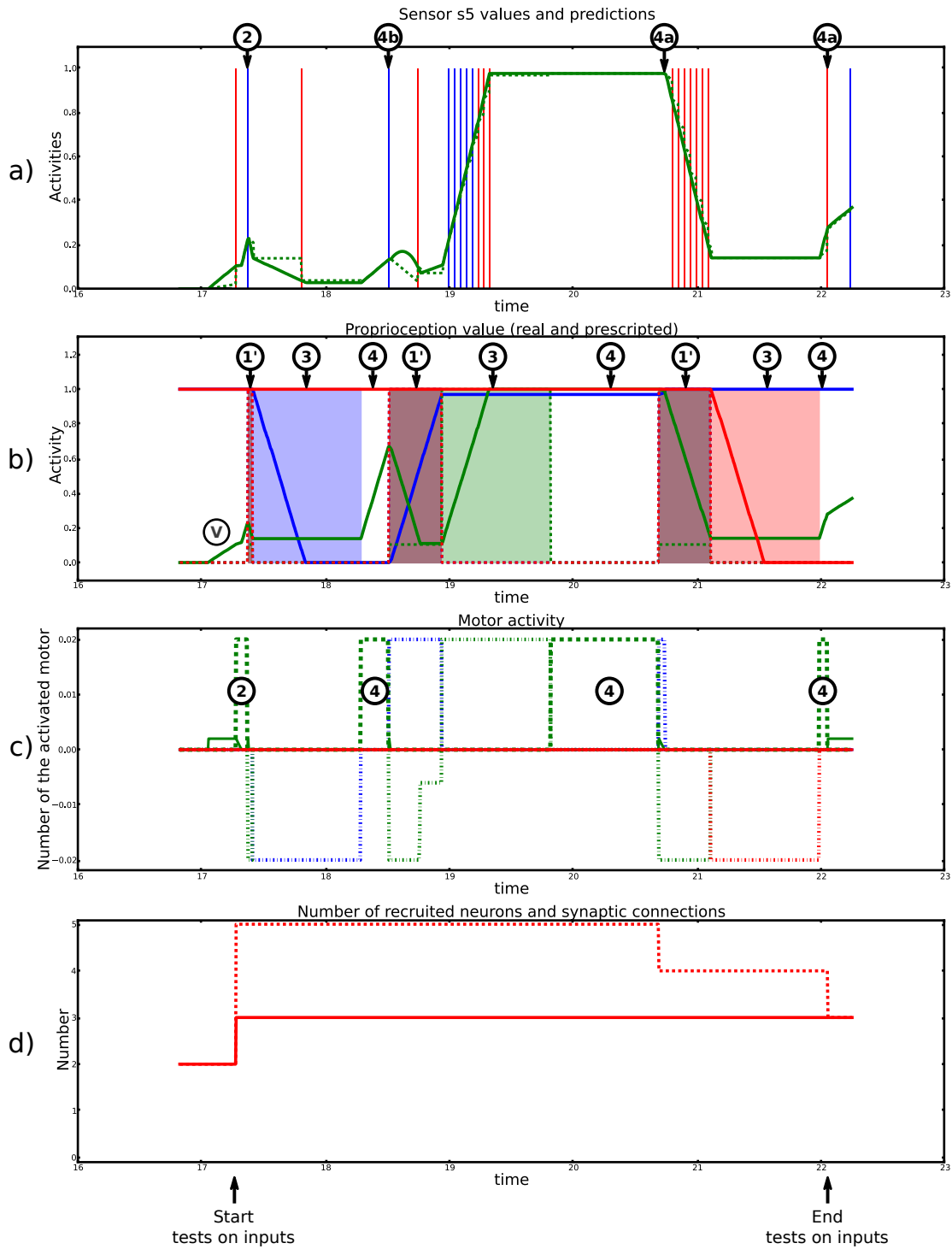


FIGURE 4.10 – Suite de la figure 4.9. On y voit les différentes étapes de recrutement lorsque plusieurs senseurs sont potentiellement pertinents pour catégoriser le nouveau changement de loi observé.

- Après un retour, toujours manuellement effectué, à $s_1 = s_5 = 0$ (observable en b)-**II**, et notons que le SLE associé au moteur négatif recrute lui aussi une catégorie), le moteur m_0 est à son tour activé, jusqu'à ce que $s_0 = 1$ (voir fig.4.9 b)-**III**). Soulignons que la loi de s_5 a désormais changé, et devient $s_5 = s_1 \cdot 1$.
- On active manuellement le moteur m_2 jusqu'à ce que $s_2 = 1$ (voir b)-**IV**). Cela ne change pas la loi, mais lorsqu'à la prochaine étape, une erreur sera détectée dans la loi apprise par le SLE_{s_5} , deux senseurs auront varié et devront être testés, dont l'un est effectivement cause du changement de loi et l'autre non.
- On active maintenant à nouveau le moteur m_1 . Puisque la loi a changé, une erreur est détectée (voir fig.4.10 b)-**V**). Une catégorie est recrutée, et de fait la loi apprise par le SLE est modifiée par cette nouvelle catégorie. Celle-ci a en entrée les senseurs qui ont suffisamment varié, et ils doivent être testés. Il s'agit ici de s_0 qui est à 1 alors que $s_0^{Stored} = 0$, de s_1 qui est à 0.104 alors que $s_1^{Stored} = 1$, et de s_2 qui est à 1 alors que $s_2^{Stored} = 0$. La figure 4.12 montre une illustration du réseau au moment du recrutement, soit avant les tests sur les entrées. Les étapes de ces tests, que nous allons maintenant décrire, sont montrées à la figure 4.10.
 - **Étape 1** : La proprioception étant déjà dans les positions s^1 , il ne se passe rien.
 - **Étape 2** : Après l'activation du moteur m_1 , la prédiction est correcte, on garde donc la catégorie (voir courbes c) puis a)).
 - **Étape 1'** : Dans la zone violette (mélangeant les couleurs bleue, verte et rouge), chaque senseur revient à sa position s^1 : en ce cas uniquement s_1 doit revenir vers $s_1^1 = 0.104$ (voir courbe b).
 - **Étape 3** : Le lien concernant s_0 est testé. Le senseur s_0 est donc mis à $s_0^0 = 0$: la courbe correspondante, b) en bleu, descend. Notons qu'afin d'être sûr d'être dans une position où le lien désactive la catégorie, même au cours du mouvement qui s'en suit, dans notre implémentation nous avons ajouté un mécanisme permettant d'attribuer une valeur désirée plus éloignée encore de s^1 que ne l'est s^0 , lorsque le mouvement qui suit active la catégorie. Et donc, dans notre cas, une valeur en dessous de 0 est désirée. Or la proprioception étant bornée, il faut ici attendre le signal de *timeout*.
 - **Étape 4** : Le moteur est activé, voir courbe c).
 - **Étape 4b** : Une prédiction correcte est détectée : le lien concernant s_0 est conservé dans la liste des liens nécessaires.
 - **Étape 1'** : Dans la zone violette, chaque senseur, ici s_0 et s_1 , revient à sa position s^1 .
 - **Étape 3** : Le lien concernant s_1 est maintenant testé. Le senseur s_1 est mis à $s_1^0 = 1$, la courbe correspondante, b) en vert, monte.
 - **Étape 4** : Le moteur est activé, voir courbe c). Cependant, l'articulation correspondante s_1 ayant déjà atteint sa valeur maximale, il ne se passe rien (voir courbe b).
 - **Étape 4a** : Le *timeout* ayant des conséquences similaires à une erreur (pour éviter des recrutements abusifs), le lien concernant s_1 est retiré de la liste des senseurs nécessaires.

- **Étape 1'** : Dans la zone violette, chaque senseur revient à sa position s^1 , et donc ici s_1 retourne à la valeur 0.104.
- **Étape 3** : Le lien concernant s_2 est maintenant testé. Le senseur s_2 est mis à $s_2^0 = 0$, et l'on peut observer la courbe correspondante, b) en rouge, descendre.
- **Étape 4** : Le moteur est activé, voir la courbe c).
- **Étape 4a** : Une erreur de prédiction est détectée (voir courbe a)) : le lien concernant s_2 est retiré de la liste des senseurs nécessaires.

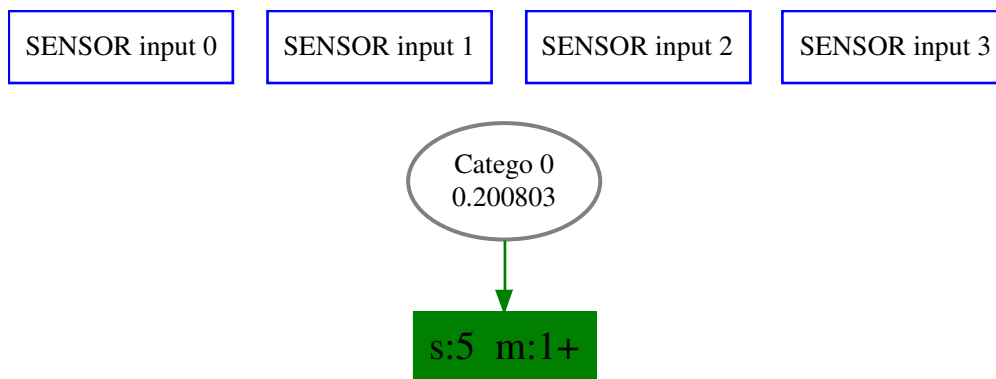


FIGURE 4.11 – Schéma représentant les catégories du SLR du moteur m_1 et du senseur s_4 , pour les mouvements positifs. Recrutement de la première catégorie : la valeur associée à la catégorie 0 (0.200803) indique le poids appris par le LMS (voir fig.4.2).

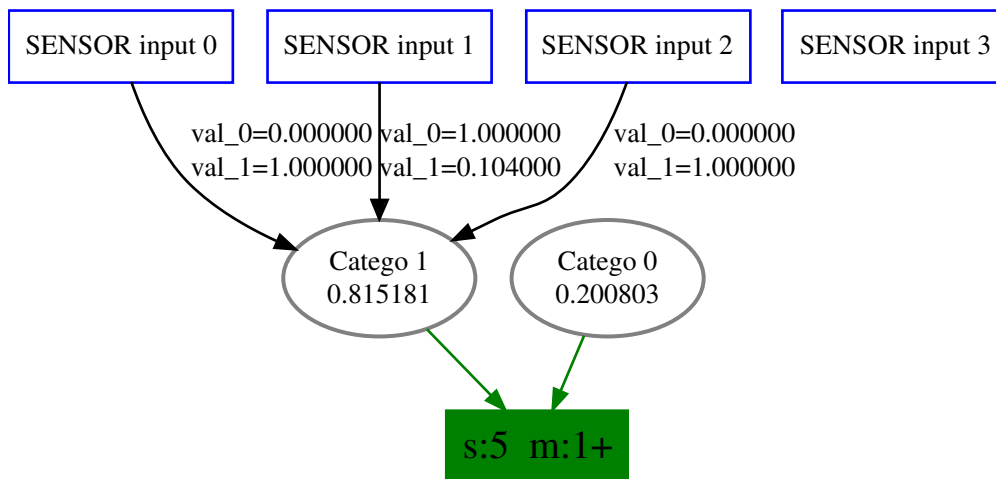


FIGURE 4.12 – Schéma représentant les catégories du SLR du moteur m_1 et du senseur s_4 , pour les mouvements positifs. Réseau au moment du recrutement de la catégorie 2. Les liens allant des “SENSOR input” à “Catego 1” indiquent les liens vers les différents senseurs, et val_0 et val_1 les valeurs s^0 et s^1 .

La figure 4.13 montre une illustration du réseau après le test sur les entrées.

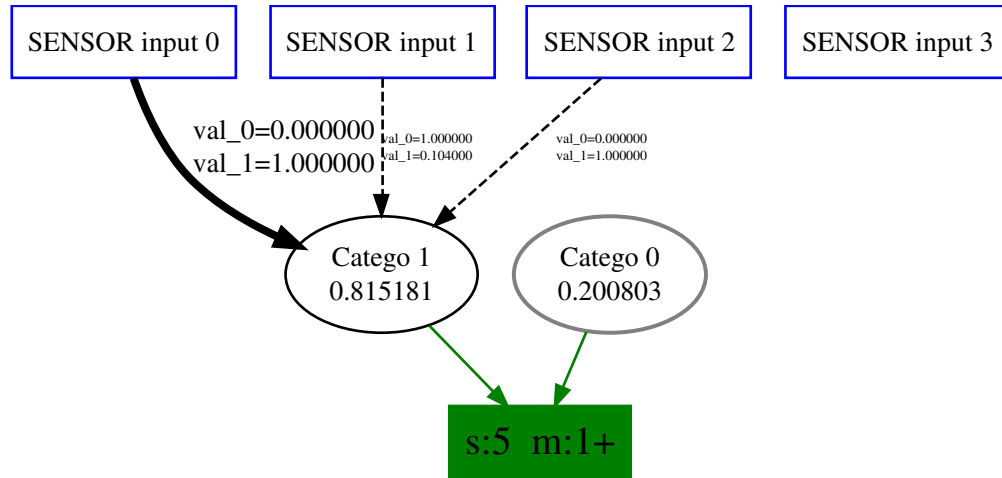


FIGURE 4.13 – Schéma représentant les catégories du SLR du moteur m_1 et du capteur s_4 , pour les mouvements positifs. Réseau à l'issue du recrutement et des tests qui s'en suivent. Les liens d'activations en gras sont conservés, et ceux en pointillés sont désactivés.

Puisque nous avons $m_1 = \dot{\theta}_1$, $s_0 = \theta_0$ et $\delta s_5 = \dot{x}^1$, ce réseau peut se traduire par l'équation suivante :

$$\dot{x}^1 = 0.200803 \cdot \dot{\theta}_1 + 0.815181 \cdot \mathcal{R} \left[\frac{\theta_0 - 0}{1 - 0} \right] \cdot \dot{\theta}_1 \quad (4.8)$$

Nous voyons donc que ce réseau constitue une approximation de l'éq.4.9, dont la précision dépend du seuil Th_1^{Err} , ici arbitrairement fixé à 0.1.

Dans cet exemple, la loi qui est globalement modifiée est linéaire durant le premier apprentissage liant s_5 à s_1 . Toutefois, elle pourrait avoir été non-linéaire comme dans l'exemple précédent. De manière similaire, cette première loi apprise est ici linéairement modifiée, après la variation de s_0 , or elle pourrait, de la même manière, être non-linéairement modifiée. En effet, chaque apprentissage apprend la frontière d'une non-linéarité en étant additif, et on peut donc ajouter autant de non-linéarités que nécessaire. Enfin, rappelons que le nombre de capteurs déterminant le changement de loi peut être aussi important que l'on veut.

Dans la partie suivante, nous allons tester les performances de l'algorithme avec une simulation du bras Katana, et comparer le nombre de liens entre les capteurs et les catégories recrutées, ainsi que les performances de cet algorithme et de celui décrit dans la section 3.1.2 basé sur une méthode appelée "absolue", et apprenant des associations entre la position de l'extrémité du bras et la proprioception.

4.3.2 Performances et limitations

Mesures effectuées

Afin de mesurer les performances de notre modèle, nous effectuerons plusieurs simulations. À chaque fois, le robot commence avec aucune connaissance préalable, et apprend ses lois sensorimotrices lors d'un babillage moteur aléatoire. Régulièrement, nous arrêterons le babillage

moteur ainsi que l'apprentissage, afin de tester les performances du système. Deux types de tests sont alors effectués.

- D'une part, une série de mouvements aléatoires est effectuée, identique à chaque test, et l'on mesure les *erreurs de prédictions* :
 - Pour DSM, cette erreur consiste en la somme des valeurs absolues des différences entre les valeurs de senseurs prédites par les SLEs, et les valeurs réelles de ces senseurs.
 - Pour le modèle absolu (ABS, voir section 3.1.2), cette erreur consiste en la somme des valeurs absolues des différences entre les configurations articulaires réelles, et celles en sortie du LMS lorsqu'il y a, en entrée du SAW, les valeurs réelles des senseurs.
- D'autre part, une tâche d'atteinte est effectuée. Celle-ci, à l'instar de celle effectuée dans la partie 3.1.4, comporte les étapes suivantes. La cible se déplace pendant un temps donné puis se stabilise afin de permettre au bras de se rapprocher autant que possible de la cible. Dans nos expériences, cette dernière varie pendant 80 itérations avant de se stabiliser pour 40 itérations. Le test d'atteinte dure lui 10000 itérations. Deux types de mesures d'erreur sont alors effectuées pour chacun des deux modèles :
 - La somme des distances cartésiennes entre la position de l'extrémité du bras et la cible à chaque itération de la tâche d'atteinte.
 - La somme des distances cartésiennes entre la position de l'extrémité du bras et la cible à la fin des 40 itérations où la cible ne bouge plus, afin d'avoir la distance finale.

Notons que ces deux erreurs ne sont pas comparables, et ne seront donc pas affichées sur les mêmes courbes.

- Pour finir, nous avons mesuré la *quantité de mouvements effectués durant l'atteinte*, en additionnant l'ensemble des variations de la proprioception durant cette tâche, pour les deux modèles.

Durant les phases de babillage, nous avons également recueilli des informations concernant le *temps passé au recrutement* et à tous les mouvements de tests qui y sont liés, et que nous appelons ici comportement de jeu. Le temps consacré à chaque activité, jeu ou babillage moteur aléatoire, ne prend pas en compte les moments où le robot effectue ses tâches d'atteintes, et est exprimé en pourcentage du temps total.

Enfin, nous avons enregistré le *nombre de catégories recrutées* par les deux modèles, afin d'afficher le nombre total de liens nécessaires. Pour DSM, lorsque qu'une nouvelle catégorie est recrutée, celle-ci est en sortie reliée au LMS du senseur du SLE concerné : il y a donc chaque fois 1 lien en sortie de la catégorie. En entrée de celle-ci, il y a un nombre variable de liens, qui dépend à la fois du nombre de senseurs ayant varié depuis la dernière bonne prédiction, et des résultats de la phase de tests du recrutement qui peut faire diminuer ce nombre. Le nombre total de ces liens N_{Liens} et le nombre de catégories N_{Cat} sont enregistrés. Ainsi, le nombre total de liens dans ce réseau est de $N_{Tot}^{DSM} = N_{Cat}^{DSM} + N_{Liens}^{DSM}$. Pour ce qui est du modèle absolu, nous

enregistrons également le nombre de catégories N_{Cat}^{ABS} . Mais, considérant que chaque catégorie est, en entrée, reliée à tous les senseurs, au nombre de N^S , et est, en sortie, reliée au LMS de la proprioception, contenant donc autant de neurones qu'il y a de moteurs N^M , le nombre total de liens est donc $N_{Tot}^{ABS} = N_{Cat}^{ABS} * (N^S + N^M)$. Puisque dans nos expériences $N^S = 3$ et $N^M = 4$, nous avons donc $N_{Tot}^{ABS} = N_{Cat}^{ABS} * 7$.

Résultats avec la simulation 2

Nous avons tout d'abord testé les performances en utilisant la simulation précédemment décrite, dans les équations 4.9. Les résultats sont montrés dans la figure 4.14.

Dans ce cas simple, où très peu de lois différentes doivent être apprises, on voit un net avantage pour DSM. En effet, celui-ci a convergé à l'itération 40 000 quand ABS continue de recruter jusqu'à l'arrêt de l'expérience, à l'itération 720 000. De plus, on constate à cette itération :

- De meilleurs scores d'atteinte (erreur accumulée à 2243.377197 pour DSM contre 2289.488525 pour ABS, et la somme des distances à l'issue de chaque atteinte est à 14.988317 pour DSM contre 15.165699 pour ABS).
- DSM a recruté moins de catégories (22 pour DSM contre 832 pour ABS). Et donc, puisqu'il y a également 22 liens recrutés, il y a donc $22 + 22 = 44$ liens au total pour DSM, et puisqu'il y a 4 moteurs et 3 senseurs, $832 * 7 = 5824$ liens au total pour ABS.

Notons enfin que le comportement de jeu est important au début, puis se stabilise assez rapidement, ce qui coïncide avec le ralentissement puis l'arrêt des recrutements.

Afin de voir le type de recrutement ayant été effectué, nous reportons dans la figure 4.15 les réseaux relatifs au senseur s_6 . Notons que celui-ci est dépendant des moteurs m_1 et m_2 , puisque $m_1 = \delta s_1$ et $m_2 = \delta s_2$, et que sa loi réelle est la suivante :

$$s_6 = s_2 \cdot (0.2 + 0.8 \cdot s_1) \quad (4.9)$$

Durant ce babillage aléatoire, nous pouvons constater que même si de nombreux liens non pertinents ont été créés au moment des recrutements, seuls les liens pertinents sont sélectionnés. En effet, lorsque m_2 est la variable, s_1 est fixe et paramétrise la loi qui s'applique lors de la variation de s_6 par m_2 . On voit ainsi dans la figure 4.15 que a) et b) ne dépendent que de s_1 . Réciproquement, lorsque m_1 est la variable, s_2 est fixe et paramétrise la loi qui s'applique lors de la variation de s_6 par m_1 . De même, on voit que dans la figure 4.15 c) et d) ne dépendent que de s_2 .

Résultats avec la simulation 3

Afin d'évaluer les performances du SLR intégré à DSM, nous avons utilisé cette fois une simulation du bras Katana (voir figure 3.5). La relation entre la position de l'extrémité du bras, dans l'espace cartésien, et des 4 articulations est donnée par les équations suivantes :

$$x = (L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)) * \cos \theta_0 \quad (4.10)$$

$$y = (L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)) * \sin \theta_0 \quad (4.11)$$

$$z = L_0 + (L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3)) \quad (4.12)$$

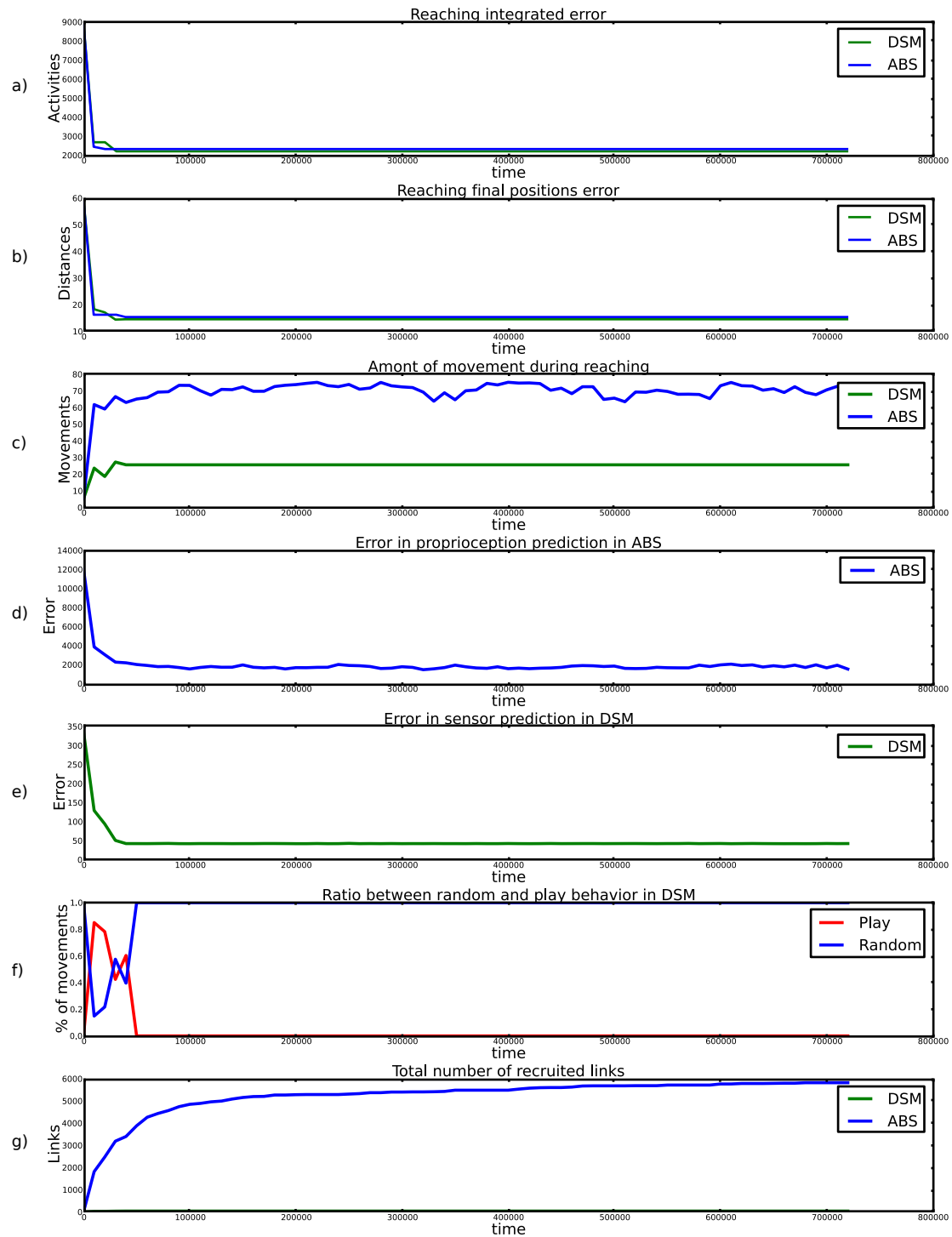


FIGURE 4.14 – Performances comparées entre DSM et le modèle absolu (ABS). a) et b) représentent respectivement l’erreur accumulée, et l’erreur par rapport à la position finale, lors de la tâche d’atteinte. c) montre la quantité de mouvements accumulés pendant les mouvements d’atteinte. d) et e) montrent l’erreur de prédiction pour DSM et ABS, respectivement. f) représente le ratio entre le comportement dit de jeu (pour le recrutement), et celui de babillage moteur aléatoire. g) montre le nombre de liens recrutés par DSM et ABS.

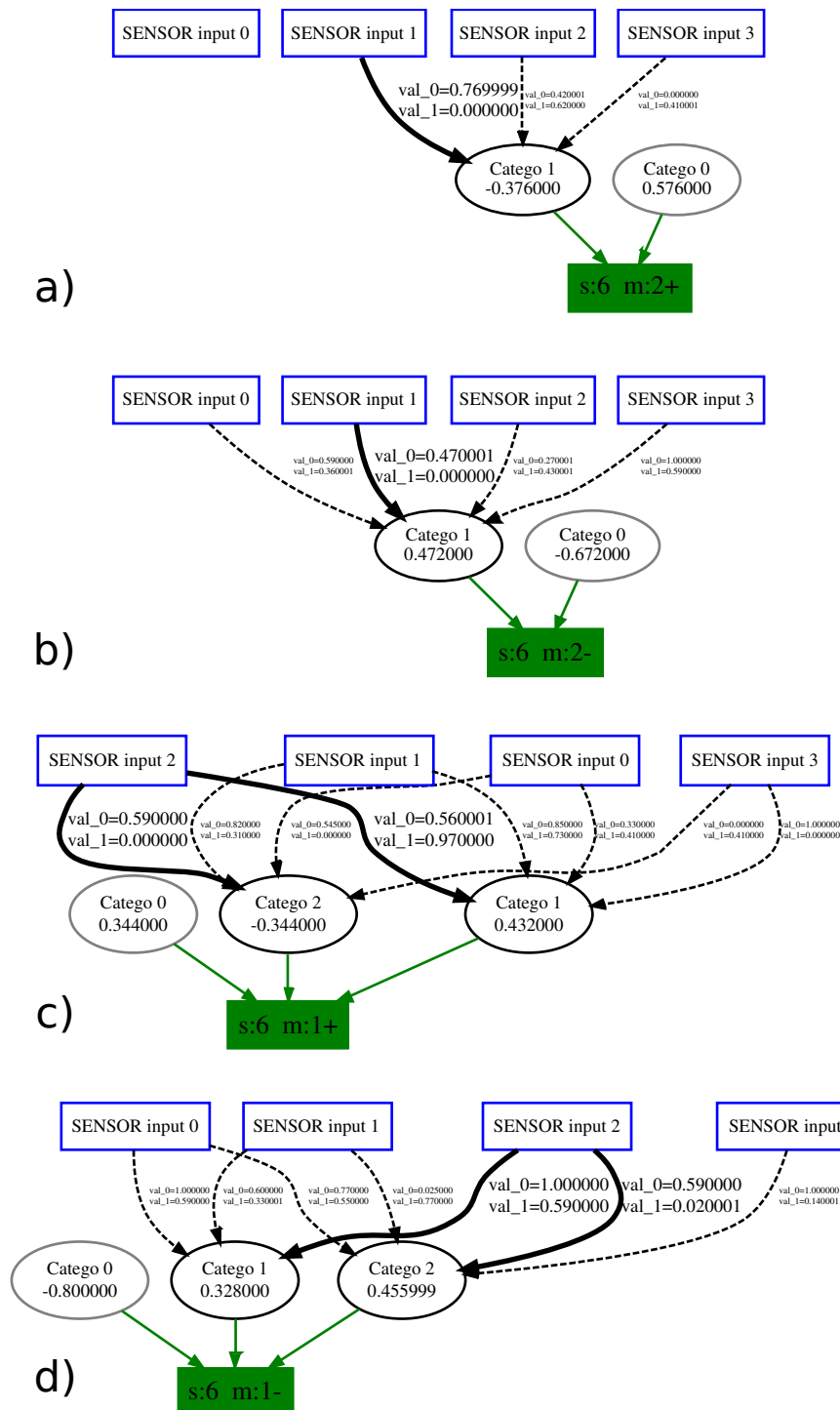


FIGURE 4.15 – Réseaux appris par le SLR concernant le capteur s_6 . Le SLE a découvert deux moteurs pouvant faire varier ce capteur : m_2 , représenté dans a) et b), et m_1 représenté dans c) et d)). Il y a de plus deux réseaux par moteur : ceux pour les grandeurs motrices positives (a) et c)), ainsi que ceux pour les grandeurs négatives (b) et d)). Comme précédemment, les liens en gras sont ceux qui après la phase de jeu ont été jugés nécessaires, et font partie du réseau final, et ceux en pointillés ont été jugés non nécessaires, qui ont été supprimés du réseau et ne figurent dans ces illustrations qu'à titre informatif. Afin de comprendre la prédiction effectuée, et donc la loi apprise, il importe donc de ne considérer que ceux en gras.

Où :

- x, y, z sont les coordonnées de l'extrémité du bras dans l'espace cartésien,
- $\theta_0, \theta_1, \theta_2, \theta_3$ sont les 4 degrés de liberté du bras Katana,
- L_0, L_1, L_2, L_3 sont les longueurs des 3 segments du robot (notons que la première articulation est éloignée de la distance L_0 du sol selon l'axe z , et qu'elle contient les 2 premiers degrés de liberté) : $L_0 = 317mm, L_1 = 190mm, L_2 = 139mm, L_3 = 315mm$.

Nous avons suivi le même protocole expérimental, avec cette fois une vigilance à 0.2 pour DSM, et à 0.98 pour ABS. Les résultats sont montrés dans la figure 4.16.

Il s'agit ici, comme précédemment, des résultats obtenus au cours d'une expérience afin d'observer le comportement de chaque modèle. L'erreur d'atteinte, ainsi que l'erreur de prédiction décroît dans cette simulation aussi relativement vite. Notons que, bien que DSM converge avec une erreur moins grande, pendant un temps l'erreur est plus importante. Ceci peut s'expliquer par le fait qu'une articulation (au moins) n'était pas sollicitée dans certaines situations, ce qui peut être dû au fait qu'aucun apprentissage n'aura été effectué sur le couple sensorimoteur associé, ou encore par le fait que suite à un recrutement, l'atteinte peut-être temporairement impactée (voir les explications sec. 4.3.1).

Comme précédemment, DSM réclame moins de quantité de mouvements pour avoir de meilleures performances. Le comportement de jeu décroît lui aussi au cours du temps, sans pour autant atteindre 0. En effet, l'apprentissage n'est ici pas parfait : on observe qu'une erreur est régulièrement détectée, mais son recrutement s'interrompt et échoue à cause de l'une des contraintes imposées par l'algorithme de recrutement. Plus précisément, bien souvent cette contrainte est le fait que l'erreur ne soit pas répétable (étape 2) : ceci arrive notamment à la frontière de deux lois, où une erreur peut être détectée sans que celle-ci ne puisse être reproduite lors du test qui suit. Les légères oscillations du comportement de jeu, une fois le recrutement terminé, s'explique par le babillage aléatoire qui provoque plus ou moins d'erreurs de ce type.

Dans la courbe g), on voit que la majeure partie des catégories sont recrutées à partir d'erreurs provenant de l'activité du moteur m_0 , celui-ci étant le moteur au pied du bras Katana et donc dont l'influence sur les variations de l'effecteur terminal est la plus importante, et dépendante de la position des autres articulations.

Notons enfin que si bien plus de catégories ont été recrutées dans cette simulation, dont la complexité est bien plus importante, il y en a toujours significativement moins que dans le modèle absolu.

Nous allons maintenant comparer les résultats de simulations en faisant varier les méta-paramètres, à savoir la vigilance utilisée dans chacun des deux modèles, et regarder la moyenne et l'erreur quadratique de chaque résultat. Dans la figure 4.17, chaque point des courbes est le résultat de 5 expériences différentes avec les mêmes paramètres. Les résultats, selon les différents critères précédemment expliqués, sont montrés pour les deux modèles (à gauche pour la méthode absolue et à droite pour DSM). La précision du modèle augmente de gauche à droite dans les deux cas : pour la méthode absolue, plus la vigilance est élevée, plus il y aura de catégories (car une vigilance élevée indique que le contexte doit être fortement similaire à l'un de ceux recrutés pour ne pas en recruter un nouveau). La vigilance est donc croissante, de 0.95 à 0.99. Pour DSM en revanche, plus la vigilance est faible, plus le nombre de catégories est élevé (car

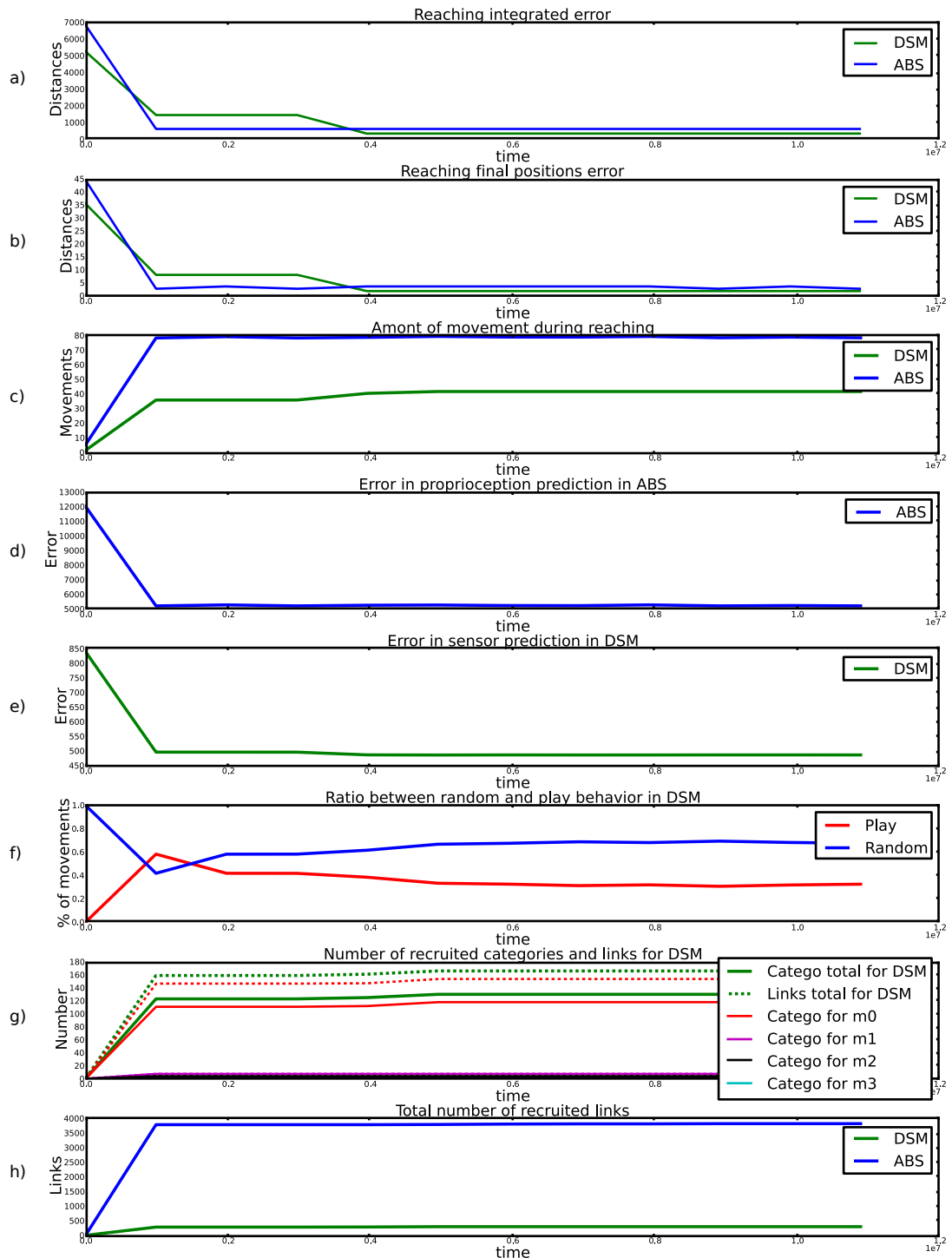


FIGURE 4.16 – Performances comparées entre DSM et le modèle absolu (ABS). a) et b) représentent respectivement l’erreur accumulée, et l’erreur par rapport à la position finale, lors de la tâche d’atteinte. c) montre la quantité de mouvements accumulés pendant les mouvements d’atteinte. d) et e) montre l’erreur de prédiction pour DSM et ABS, respectivement. f) représente le ratio entre le comportement dit de jeu (pour le recrutement), et celui de babillage moteur aléatoire. g) montre le nombre de catégories et de liens appris par DSM, et plus spécifiquement par chaque moteur h) montre le nombre de liens recrutés par DSM et ABS.

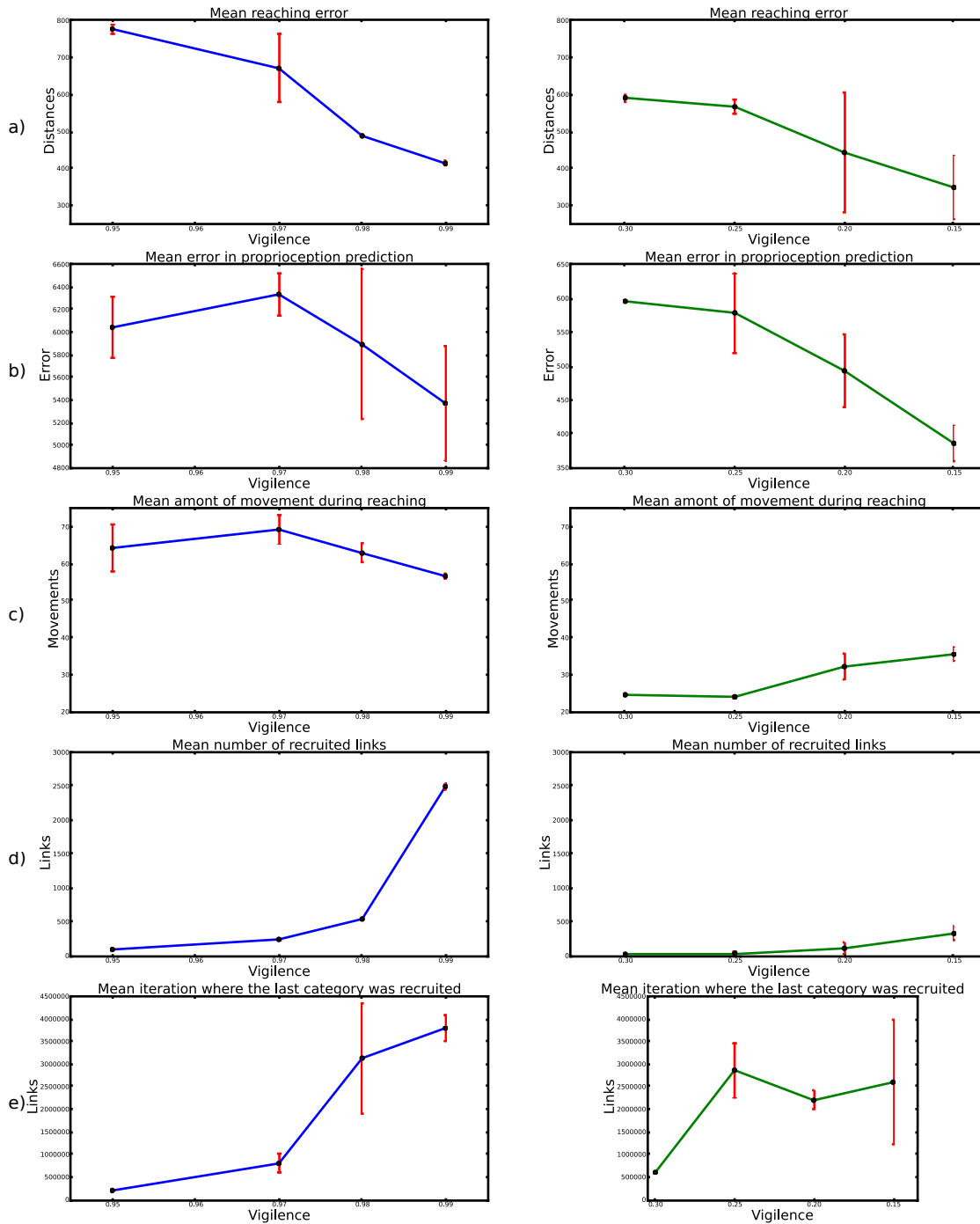


FIGURE 4.17 – Courbes des résultats par vigilance, pour la méthode absolue (colonne de gauche) et pour DSM (colonne de droite), après 5 000 000 d’itérations. a) indique l’erreur moyenne d’atteinte, b) l’erreur moyenne de prédiction, c) la quantité de mouvement, d) le nombre le liens recrutés, et f) l’itération moyenne à laquelle la dernière catégorie a été recrutée. Notons qu’excepté la courbe b), les axes des ordonnées sont les mêmes et donc les valeurs sont comparables.

une vigilance faible indique une moins grande tolérance aux erreurs de prédiction). La vigilance est donc décroissante, de 0.3 à 0.15.

Sans surprise, les erreurs d'atteinte et de prédictions diminuent à mesure que la vigilance augmente. De la même manière, le nombre de liens recrutés augmente. La courbe e) donne, elle, une nouvelle information, à savoir le moment du recrutement de la dernière catégorie, ce qui peut donner une appréciation empirique de la convergence de chaque système. Le problème étant que le mode de parcours est aléatoire, et que la combinatoire (très grande) croît à mesure que la vigilance augmente, et d'autant plus pour DSM qui doit explorer les directions de mouvements possibles à partir de chaque position, et non les seules positions comme pour la méthode absolue. Cependant, la méthode semble empiriquement converger, dans la mesure où de manière répétable aucune catégorie n'est recrutée pendant un temps relativement long, à chaque expérience.

4.4 Discussion et conclusion

4.4.1 Remarques générales

Nos travaux, se concentrant sur une approche développementale sensorimotrice, ne sont donc pas centrées sur un apprentissage de machines traditionnel. De fait, de nombreuses pistes et comparaisons pertinentes avec d'autres travaux restent encore à développer. Mais nous insistons sur le fait que cette approche ne vise pas tant la performance, que notamment une capacité à apprendre en ligne et de manière incrémentale, ou encore l'apprentissage de lois sensorimotrices exploitables pour agir et effectuer des séquences d'actions. Notre approche exploite ainsi le parcours, l'historique du domaine sensorimoteur parcouru, en inférant des lois entre situations passées et présentes. Il utilise de plus l'action pour une exploration active de l'environnement et pour tester des hypothèses liées à son exploration. Cette exploration des hypothèses repose sur le même mécanisme que celui décrit dans la partie précédente, permettant de faire des séquences. De fait, le même mécanisme peut-être utilisé tant pour satisfaire des tâches émanant d'une motivation extrinsèque, que pour l'apprentissage même des lois sensorimotrices, le SLR produisant ainsi des motivations intrinsèques que l'on peut rapprocher d'un comportement de jeu [Oudeyer et al., 2007; Lee, 2011].

4.4.2 Densité du réseau, et création d'unités sensorielles et motrices

Le faible nombre de liens recrutés, lors de la dernière expérience, traduit un faible coût computationnel du système, en opposition avec la lourdeur que l'on pourrait présumer, lorsque l'on considère que chaque couple de senseur et moteur fait l'objet d'un apprentissage indépendant, chacun prenant en entrée tous les senseurs et tous les moteurs possibles. Ainsi, peu importe le nombre de senseurs ou de moteurs, et peu importe également si leur nombre augmente au cours de l'expérience, la taille effective du réseau ne sera théoriquement fonction que des covariations observées, que, par ailleurs, celles-ci soient liées aux lois réelles qui s'appliquent au système apprenant, ou aux simples contingences expérimentales. Ce faible coût computationnel est bien sûr à mettre en perspective avec le temps d'apprentissage qui, lui, est long, du fait du comportement de jeu.

Une limitation apparaît toutefois lors de cas extrêmes. Si, par exemple, des centaines de senseurs sont utilisés (ce qui est imaginable si ces senseurs sont très bas niveau, ou si l'on imagine un cas extrême où chaque pixel est considéré comme un senseur). Dès lors tester chaque entrée devient non seulement excessivement long, mais est, de plus, impossible en pratique puisqu'il est impossible de faire varier un unique pixel : des lois physiques, ou sensorimotrices, relient en effet ces senseurs entre eux, et chez l'humain des traitements très bas niveaux sont effectués dans l'encodage même de l'information visuelle. Cet exemple, même s'il peut sembler absurde du point de vue d'un modélisateur, met en évidence ce qui doit constituer la prochaine étape de réflexion : un travail sur ce que sont, ou doivent être, ces senseurs, et d'une même manière, ce que sont ces moteurs, que l'on devrait trouver en entrée de DSM.

Car les senseurs, de même que les moteurs, sont également couplés entre eux. Avec ces considérations, il faut également souligner un autre problème, qui est en partie lié : il est possible que la variation d'une loi soit due à la combinaison (qui peut être arbitrairement complexe) entre les différentes variations de senseurs. Ainsi, l'algorithme de jeu que nous avons proposé devrait théoriquement prendre en compte cette combinatoire. Toutefois, une première manière

de chercher des solutions à ce problème consiste à repenser les entrées des SLEs, en s'inspirant par exemple de la neurobiologie. Il pourrait s'agir de primitives comme des synergies motrices (ou des DMP, mouvements dynamiques primitifs, voir [Ijspeert et al., 2013]), ainsi que des entrées sensorielles issues de pré-traitements, ce qui permettrait déjà de réduire fortement la dimensionnalité d'entrée. Ces solutions sont évidemment à prendre en compte, mais notons que l'approche et l'efficacité de DSM résident justement dans le fait que le modèle dispose et traite d'entrées motrices et sensorielles aussi atomisées que possible. Le modélisateur est donc de nouveau confronté à un choix difficile, consistant cette fois non à déterminer quelles entrées fournir, problème dont a traité ce chapitre, mais quelle granularité leur donner.

Comme évoqué dans la section 3.2.5, nous pourrions utiliser les possibilités de construction hiérarchique de DSM, lequel permet, comme dans [Chaput, 2004], aux couches supérieures d'avoir pour entrée la sortie d'une couche inférieure.

Mais en dehors de primitives, ou d'autres pré-traitements (généralement codés en dur), la problématique générale reste celle d'unifier certains senseurs entre eux, et certains moteurs entre eux. Pour cela, nous pensons qu'il faudrait travailler l'idée consistant à considérer la boucle sensorimotrice elle-même comme étant génératrice d'unité. On peut, en effet, imaginer que la diversité de ce qui varie, aussi bien dans ce qui est observé (sensoriel) que généré par le robot (moteur), ou encore le rapport entre les deux (lois), puisse être unifiée par la saisie commune des conséquences de l'action. Soulignons que la constitution de telles unités perceptuelles tirerait sa légitimité du fait qu'elles sont postérieures à l'expérience sensorimotrice. Dans cette hypothèse, nous pensons que DSM pourrait être utilisé pour arriver à cette fin, étant donné qu'il est entièrement dédié à l'apprentissage, et à l'expérimentation de lois régissant justement ce qui varie (les senseurs, ou les lois elles-mêmes). Dans une première approche par exemple, on pourrait imaginer unifier, en un senseur composé, tous les effets de variations observées sur les senseurs d'un unique moteur.

Notons que de telles unités nécessiteraient alors, pour être utilisées au sein de DSM, des règles permettant l'adoption de mesures de distances applicables à des espaces de multiples dimensions, qu'elles soient sensorielles, ou motrices, ce qui n'est pour l'instant pas présent dans le modèle.

4.4.3 Processus maturationnel et exploration active

Le recrutement de lois additives a des conséquences intéressantes d'un point de vue développemental. En effet, rappelons que, d'une part, chaque apprentissage n'a de sens que par rapport aux apprentissages passés, et que, d'autre part, ces premiers apprentissages sont, dans l'état actuel du modèle, dépendants de la première erreur qui est constatée, laquelle dépend en partie du seuil d'erreur et en partie de l'aléatoire du babillage moteur effectué.

De ce fait, et comme nous l'avons vu au cours de nos expériences, l'ordre de ce qui est appris a une importance non négligeable sur ce qui est appris et encodé. Or, cet ordre pourrait être influencé par des contraintes maturationnelles, en reproduisant des contraintes physiologiques telles qu'observées notamment en psychologie et utilisées en robotique développementale (voir [Berthier et al., 1999; Berthouze and Lungarella, 2004; Lee et al., 2007; Baranes and Oudeyer, 2011]). Une seconde manière d'influencer cet ordre serait d'orienter l'apprentissage par une exploration active. Ici, la valeur d'affordance apprise (voir sec. 3.2.2), qui constitue en quelque sorte l'efficacité d'un moteur dans le contexte courant, peut jouer un rôle. Si, par

exemple, on biaise l'exploration en activant prioritairement les moteurs ayant une grande efficacité (c'est-à-dire où une faible activité motrice entraîne une grande variation sensorielle), alors on peut s'attendre à ce que le robot se concentre d'abord sur les articulations proximales, puis distales, reproduisant en partie le type de résultats obtenus dans [Stulp and Oudeyer, 2012]. On pourrait également imaginer des critères plus élaborés, prenant notamment en compte le nombre de catégories ou liens recrutés afin de se concentrer en priorité sur les moteurs des articulations les plus dépendantes du "contexte" courant. Car on pourrait alors espérer une généralisation plus simple et efficace lors de l'altération globale de ces lois par les autres contextes (en l'occurrence les autres articulations). Par exemple, dans le cas où une seule articulation en mouvement génère de nombreuses lois, il semble intéressant de commencer par apprendre ces seules lois, en n'activant que cette articulation, avant d'effectuer d'autres mouvements. On s'attend alors à ce que ceux-ci aient une influence globale plus efficace sur toutes ces précédentes lois (voir également section 4.4.4). Enfin, il serait intéressant de privilégier l'apprentissage des senseurs qui contextualisent le plus les autres lois. En effet, maîtriser une loi sensorimotrice signifie aussi bien apprendre les relations entre moteurs senseurs, selon leurs contextes, que, justement, être à même d'agir sur ces contextes, et donc d'être capable d'agir sur ses propres lois.

4.4.4 Limitations et améliorations possibles

Tout d'abord, nous n'avons pas de preuve de convergence de notre algorithme, seulement des résultats statistiques expérimentaux probants.

Par ailleurs, au vu du fonctionnement de cet algorithme qui procède par correction, en ajoutant des transformations linéaires globales, on imagine que le cas idéal d'application est lorsque de telles modifications sont mathématiquement pertinentes. Or, nous pouvons imaginer que, de par son éventuelle propriété d'indépendance, lorsqu'une modalité "fait contexte" sur un ensemble en apparence indépendant de lois sensorimotrices, la situation soit mathématiquement plus favorable. Pour illustrer ceci, rappelons-nous l'idée du chaînage des neurones à champs de gain (section 2.2) : il s'agissait alors de chaîner les modalités les unes aux autres (voir figure 2.8) afin, qu'à terme, la chaîne causale des modalités, par exemple celle pour les mouvements d'atteintes (voir [Pouget and Snyder, 1997]), dont nous nous sommes inspirés pour le modèle de la section 2.2), soit encodée et puisse être utilisée. Nous avons souligné le problème lié à la définition *a priori* de tels chaînages, et avons proposé avec DSM que seule l'exploration sensorimotrice en soit à l'origine. Mais dans notre proposition, nous pouvons de plus faire la remarque suivante. Si l'on considère le cou du robot (et les caméras) fixes durant l'apprentissage des mouvements de la main, cet apprentissage se trouvera basé sur la seule proprioception, et formera un ensemble relativement complexe de lois, comme en témoignent d'ailleurs nos expériences. Si, après cela, le cou est autorisé à effectuer des mouvements de roulis de gauche à droite, cela entraîne une rotation visuelle de ce qui est capté par la caméra. De fait, ce sera l'ensemble des lois apprises qui deviendront caduques, quand une simple transformation linéaire globale suffit ici à apprendre la transformation nécessaire. Cet exemple n'est d'ailleurs pas sans rappeler l'expérience d'ajout d'outil à un système d'apprentissage utilisant une stratégie absolue (voir 2.1.1).

Notons donc que même si ce modèle a été pensé pour ce type de problématiques, il est intéressant de voir qu'il se comporte déjà convenablement lors de situations moins idéales, comme c'est le cas avec la simulation du bras Katana, et l'utilisation de coordonnées cartésienne (et

non cylindrique), dans lequel il y a de nombreuses interactions non linéaires (voir éq.4.10), et locales.

Le but de ce chapitre était de tester les idées à la base du SLR, telles que présentées dans la section 4.1. De fait, le modèle présenté dans ce chapitre n'a pour but que d'implémenter le plus simplement ces idées, il est algorithmique et de nombreux choix ou mécanismes sont discutables, et pourraient faire l'objet d'améliorations. Nous allons en énumérer ici quelques-unes.

Le mécanisme de pondération des activités motrices pourrait être amélioré, en ne considérant plus seulement l'efficacité mais de nombreux autres critères, et l'on pourra trouver de nombreux exemples dans la littérature concernant le contrôle optimal (voir 1.2.2), ce qui permettrait notamment de s'approcher de certaines caractéristiques importantes des trajectoires de mouvements humains.

Concernant le mécanisme de simulation, la proposition consistant à générer du bruit sur les entrées sensorielles afin de tester des configurations aléatoires (voir section 3.2), pourrait être grandement améliorée. Grâce à la propriété du SLR en effet, chaque catégorie correspond désormais à une inflexion particulière de la loi sensorimotrice qui s'applique. Il devient donc possible d'orienter la recherche d'une configuration alternative pour que celle-ci soit guidée par l'effet recherché. De plus, chaque catégorie possède des lois d'activation et de désactivation accessibles par le robot, et donc sur lesquelles il lui est possible de jouer. Il devient dès lors possible d'envisager un mécanisme permettant de maximiser l'efficacité de la recherche de contextes alternatifs, tout en minimisant en même temps les variations sensorielles qui devront être effectuées, lorsqu'un contexte est désiré.

Une amélioration évidente du modèle consiste également à ne pas apprendre en une fois les catégories, ne serait-ce que parce que l'apprentissage d'une mauvaise catégorie a un impact global, et donc potentiellement important, mais également parce que cela permettrait un apprentissage moyenné, et donc plus fin, et plus assuré, d'une transformation. De plus, cela permettrait d'apprendre plus ou moins, selon la proximité du contexte avec celui réclamé par l'algorithme du jeu que nous avons présenté. Non seulement un apprentissage passif serait alors possible (si par hasard la situation à tester se trouve expérimentée), mais aussi cela permettrait de mieux gérer les cas où les lois sensorimotrices ne permettent pas d'atteindre un contexte particulier (parce que l'apprentissage n'est pas assez mature, ou parce que des moteurs, ou senseurs sont couplés). Notons d'ailleurs que des contextes nouveaux, et peut-être impossibles à obtenir en réalité, peuvent être désirés par le robot, lorsque la combinatoire des entrées à tester (celles ayant variées) est explorée.

Concernant toujours le recrutement d'une catégorie, dans notre modèle lorsqu'une erreur est constatée, celle-ci se traduit systématiquement par une addition, visant à corriger l'ensemble des précédents apprentissages. Le fait de pouvoir ajouter autant de catégories correctives que nécessaire, sur l'ensemble des précédents apprentissages tout en tentant de conserver ceux-ci, correspond à l'aspect incrémental de notre modèle. Le fait que ces corrections soit additives n'est qu'une proposition permettant de prendre en compte l'indépendance des entrées (voir sec. 4.1.1), en permettant d'ajouter les contributions de chaque contexte les uns aux autres. Nous avons toutefois testé notre modèle dans une situation de dépendance des entrées, et même dans cette situation, les performances ont été bonnes.

Mais, dans une perspective toujours incrémentale mais non additive, on pourrait imaginer qu'une nouvelle catégorie vienne pondérer les précédentes. Une correction pourrait être apportée

non à l'ensemble des précédents apprentissage, mais seulement à une précédente correction, ou à un sous-ensemble de corrections. Pour cela, nous pourrions envisager qu'une correction puisse se traduire non par une addition à l'ensemble des contributions, mais par une multiplication à l'une des contributions. Cela pourrait permettre des corrections plus ciblées, et donc précises et économes, mais cela poserait alors le problème du choix possible entre corrections additives et multiplicatives. Cependant, dans le cadre d'un apprentissage ne s'effectuant pas en une fois, cela pourrait se résoudre par un mécanisme statistique, permettant, par exemple, d'attribuer la correction la plus constante (par une comparaison des erreurs quadratiques des corrections, lorsque celles-ci sont effectuées de manière additive ou multiplicative, par exemple).

Enfin, l'aspect temporel pourrait être intégré. Lors d'événements ayant une dynamique propre, l'emprunte temporelle (estimée par exemple par des mesures telles que la moyenne des périodes entre deux événements) peut se révéler être un critère à prendre en compte parmi d'autres, le plaçant ainsi à égalité avec les autres entrées sensorielles qui peuvent éventuellement "faire contexte".

Pour finir, l'expérience faite au chapitre précédent, avec utilisation d'outils (section 3.2.4), devrait être refaite, avec cette fois tous les senseurs et les moteurs en entrée des SLEs, afin de voir dans quelle mesure un résultat comparable pourrait être obtenu grâce au SLR.

Conclusion

Dans cette thèse, nous nous sommes interrogés sur les possibilités offertes par différents types d'encodages sensorimoteurs. Au cours du second chapitre, nous avons expérimenté plusieurs propositions, notamment du point de vue de leur capacité à utiliser des outils, et elles ont chacune mise à jour un ensemble de limitations. Le modèle proposé par la suite, DSM, a été conçu pour répondre à ces limitations. De fait, nous avons dans le troisième chapitre présenté ce modèle, et la manière dont il permet un apprentissage de lois sensorimotrices susceptibles de réaliser des tâches d'atteintes. Par la suite, en autorisant l'apprentissage de lois dédiées à l'encodage de la variation de la position perçue de la main lors de la saisie d'un outil, nous avons montré comment ce modèle permettait également la réalisation d'une séquence d'actions non apprise, visant à saisir un outil pour atteindre un point, lorsque celui-ci est autrement hors de portée. Notons que cette loi illustre le mécanisme d'extension du schéma corporel, dont la genèse aura de fait été moins étudiée que ses effets. L'un des intérêts du modèle réside dans le fait que seules les lois sensorimotrices déterminent les actions à effectuer. Enfin, dans le dernier chapitre, nous avons proposé une méthode permettant un apprentissage plus fin, économe et incrémental de ces lois, utilisant pour cela un comportement spécifique. Sont attendues, mais cela devra être expérimenté dans de futurs travaux, des actions mieux ciblées, économes et permettant donc une meilleure généralisation, lors de l'accomplissement d'actions ou séquence d'actions effectuées dans le cadre de comportements orientés par un but.

Rappelons que puisque DSM est un modèle permettant justement des comportements orientés par un but, il n'est pas à mettre en compétition, mais plutôt en parallèle, avec d'autres modèles permettant d'autres types d'apprentissages, comme l'apprentissage par renforcement, ou encore l'approche PerAc. De la même manière, la question des motivations, intrinsèques ou extrinsèques, est parallèle à ce modèle et devra également permettre d'enrichir le comportement global du robot.

Puisque l'ensemble de la thèse a interrogé, puis proposé un modèle rendant compte de lois sensorimotrices telles qu'encodées par un robot, et donnant lieu à des actions ou séquences d'actions en vue d'une fin, nous proposons ici un rapide récapitulatif théorique de l'interprétation, portée par cette thèse, de ce qu'est une telle loi. Notons qu'il s'agit donc d'une proposition purement théorique, inspirée par nos travaux (et bien entendu, ceux que nous avons cités).

Notre première question aura donc consisté à déterminer ce qui doit être appris. Le principe que nous avons adopté, quant à l'utilisation d'outil, consiste à considérer qu'un outil doit pouvoir être perçu comme une extension de corps, quand bien même pour un observateur extérieur celui-ci n'appartient objectivement pas au corps du robot. Ce principe entraîne donc que ce qui

constitue le corps du robot, du point de vue du robot, est une chose qui doit être apprise. Le robot ne doit donc pas avoir d'*a priori* sur ce qui appartient à l'environnement, ou à son propre corps.

La schématisation de la figure 4.18 illustre cette vision. Les entrées d'un robot sont ses senseurs, mais ce qui est senti peut venir tout autant de l'environnement (comme la vision d'un outil) que du robot lui-même (comme la proprioception). De la même manière, les actions peuvent agir sur l'environnement (de par les lois physiques, ou les interactions sociales) ou bien agir sur son propre corps, voire constituer des actions que l'on pourrait qualifier de plus haut niveau. Par exemple une co-contraction musculaire peut entraîner un mouvement du bras dans une certaine direction, et un tel mouvement peut être perçu comme une action du robot. L'important, dans ce schéma, est le fait que du point de vue du robot, rien ne distingue a priori le corps du robot du monde extérieur. Plus encore, ce schéma montre que même lorsque le robot active ses propres moteurs, il ne peut savoir en quoi ceux-ci consistent. On pourrait donc presque dire qu'il ne sait pas en quoi *lui-même* consiste. L'hypothèse proposée est donc que pour un robot, *l'apprentissage sensorimoteur consiste à apprendre ce qu'il est*. Et puisque cet apprentissage n'est contraint par aucun *a priori*, celui-ci peut potentiellement excéder les limites de son propre corps. Notons que la notion d'autopoïèse (voir l'introduction, et [Varela et al., 1974]), laquelle comprend notamment l'idée d'un jeu permanent entre soi et l'environnement visant à redéfinir constamment la frontière entre les deux, semble congruente avec l'idée que nous défendons ici, à savoir que l'une des activités primordiales du robot doit être de se définir lui-même, bien qu'ici il s'agisse d'un processus cognitif. Précisons que l'absence d'*a priori* du robot signifie que, d'un point de vue théorique, avant toute expérience sensorimotrice, sa définition de lui-même est vide. De fait, sa dynamique sera de se constituer et de s'étendre, en repoussant peu à peu la frontière de sa propre perception de ce qu'il est à son corps, voire à l'environnement dans le cas de l'utilisation d'outil.

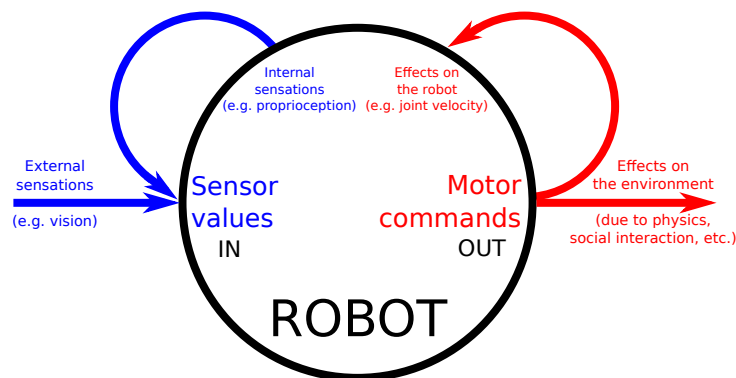


FIGURE 4.18 – Boucles de sensations et actions d'un robot.

Ainsi cette figure met en évidence le fait que le robot n'a pas de sensations directes de ce qu'il est (ou du moins ne peut savoir quand c'est le cas), ni de ce qu'il peut faire, ou même, et c'est essentiel, de ce que sont ses moteurs (c'est-à-dire ce qu'ils permettent de mettre en mouvement). Toutefois, le robot est à même de savoir qu'il actionne ses moteurs, et donc qu'il bouge. La solution, de fait, pour savoir ce qu'il est, consiste pour lui à considérer qu'il est ce qu'il peut bouger, et plus précisément à voir et à considérer ce qui est bougé comme étant ses propres moteurs. Le robot se voit ainsi comme un ensemble de moteurs, ensemble qu'il est lui-même apte à définir au cours de son développement sensorimoteur.

L'apprentissage de soi passe donc par la boucle sensorimotrice : ce que le robot sait pouvoir bouger (c'est-à-dire là où il maîtrise les lois sensorimotrices) est considéré comme lui-même. Par conséquent, ce peut être indépendant de son propre corps (comme l'illustre le fait qu'il puisse percevoir un outil comme étant une extension de son propre corps).

Il s'agit donc en quelque sorte d'un processus d'assimilation de ce qui est extérieur (au sens de capté par les senseurs) en perception de soi, en tant que ce soi est un acteur. Autrement dit il s'agit d'un ensemble de moteurs, et d'une aptitude, ou maîtrise (voire, peut-être, d'une conscience) de sa capacité à activer ses moteurs, que ceux-ci soient physiquement ceux du robot, ou extérieur à celui-ci. Soulignons que vu de cette manière, tout a priori dans un modèle sur ce qu'est le robot, est, de fait, une limitation fondamentale en ce que ça constitue un frein à ce processus d'assimilation.

Ainsi, tout comme pour l'utilisation d'outils, les sensations provenant de la main doivent pouvoir devenir une perception de soi du robot, c'est-à-dire être perçues en tant que l'un de ses moteurs : à savoir quelque chose d'"activable", sous conditions (conditions que nous avons appelées contexte, dans nos travaux).

Revenons maintenant sur l'existence de ces conditions à l'action. Dans la mesure où la manière dont il se perçoit lui-même, est en tant que *capacité à agir*, ces conditions à l'action se présentent, pour le robot, comme des conditions lui permettant d'instancier ce qu'il est.

Dans cette logique, actualiser ces contextes permet d'être un soi différent : or cette capacité même à actualiser un contexte (illustrée dans nos travaux par la réalisation de séquences) est tout autant constitutive du robot que n'importe quelle autre capacité à agir. Il vient que le robot se définit par la maîtrise d'une capacité à agir sur l'environnement, comme sur lui-même.

Pour illustrer cela, prenons l'exemple d'un bras doté de 2 degrés de liberté, figure 4.19. Selon la position de θ_2 , la loi reliant les variations de θ_1 à la position X du bras sera différente. De fait, une loi relie les variations de θ_2 à une autre loi, à savoir celle reliant les variations de θ_1 à la position X .

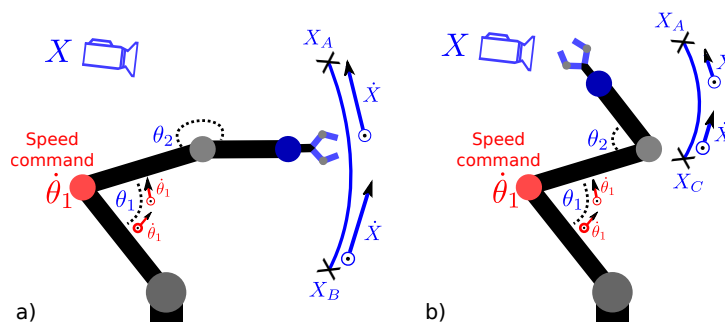


FIGURE 4.19 – Bras à 2 degrés de libertés illustrant l'influence entre elles de deux lois sensorimotrices : selon la position de θ_2 , différente en a) et en b), la loi reliant les variations de θ_1 à la position X du bras sera différente.

Nous proposons donc qu'il existe 2 types de lois sensorimotrices, comme illustré figure 4.20 :

- Des lois sensorimotrices du premier ordre (SML-1), concernant la manière dont l'activité motrice du robot change l'environnement

- Des lois sensorimotrices du second ordre (SML-2), concernant la manière dont l'activité motrice du robot change la loi de sa relation à l'environnement, c'est-à-dire change les lois du premier ordre

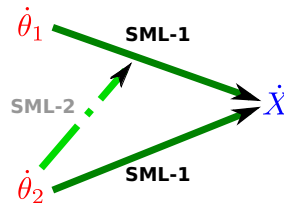


FIGURE 4.20 – Lois sensorimotrices du premier et du second ordre (respectivement en traits pleins, et en pointillé).

L'apprentissage, qui s'intéresse à ce qui est bougé (et le met en jeu), s'intéresse donc tant aux senseurs qu'aux lois sensorimotrices elles-mêmes. Dans le cas des SML-1, l'apprentissage de soi qui est fait concerne sa capacité à changer immédiatement l'environnement. Dans le cas des SML-2, l'apprentissage de soi qui est fait concerne la manière dont le robot change sa relation au monde (ses SML-1) lorsque l'environnement varie, et donc comment le changement de l'environnement change le robot lui-même.

Ces lois sensorimotrices servent tant à apprendre ce qu'est le robot, que ce qu'il peut être. Dans cette vision, l'accomplissement d'une tâche consiste pour le robot à s'utiliser comme outil pour parvenir à un état de l'environnement donné, et donc à perturber l'environnement (au sens large) pour, soit, arriver immédiatement à ce but, soit, par cette médiation, se changer lui-même et devenir apte à arriver à ses fins.

Bibliographie

- Amiaz, T., Lubetzky, E., and Kiryati, N. (2007). Coarse to over-fine optical flow estimation. *Pattern recognition*, 40(9) :2496–2503. 66
- Andersen, R. (1997). Multimodal integration for the representation of space in the posterior parietal cortex. *Philos Trans R Soc Lond B Biol Sci*, 353 :1421–1428. 55
- Andersen, R., Essick, G., and Siegel, R. (1985). Encoding of spatial location by posterior parietal neurons. *Science*, 230 :450–458. 55
- Andersen, R. and Mountcastle, V. (1983). The influence of the angle of gaze upon the excitability of the light-sensitive neurons of the posterior parietal cortex. *J. Neuroscience*, 3 :532–548. 55
- Arbib, M. A. (2011). From mirror neurons to complex imitation in the evolution of language and tool use*. *Annual Review of Anthropology*, 40 :257–273. 16
- Arbib, M. A., Bonaiuto, J. B., Jacobs, S., and Frey, S. H. (2009). Tool use and the distalization of the end-effector. *Psychological Research PRPF*, 73(4) :441–462. 15
- Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., Ogino, M., and Yoshida, C. (2009). Cognitive developmental robotics : A survey. *IEEE Transactions on Autonomous Mental Development*, 1(1) :12–34. 44
- Aschersleben, G. (2006). Early development of action control. *Psychology Science*, 48(4) :405. 21
- Bach-y Rita, P., Collins, C. C., Saunders, F. A., White, B., and Scadden, L. (1969). Vision substitution by tactile image projection. *Nature*, 221(5184) :963–964. 120
- Baillieul, J., Hollerbach, J., and Brockett, R. (1984). Programming and control of kinematically redundant manipulators. In *Decision and Control, 1984. The 23rd IEEE Conference on*, pages 768–774. 30
- Banquet, J. P., Gaussier, P., Dreher, J. C., Joulain, C., Revel, A., and Günther, W. (1997). Space-time, order, and hierarchy in fronto-hippocampal system : A neural basis of personality. *Advances in Psychology*, 124 :123–189. 39, 40
- Banquet, J.-P., Gaussier, P., Quoy, M., Revel, A., and Burnod, Y. (2005). A hierarchy of associations in hippocampo-cortical systems : cognitive maps and navigation strategies. *Neural Computation*, 17(6) :1339–1384. 40

- Baranes, A. and Oudeyer, P.-Y. (2011). The interaction of maturational constraints and intrinsic motivations in active motor development. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2, pages 1–8. IEEE. 162
- Baranes, A. and Oudeyer, P.-Y. (2013). Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1) :49–73. 111
- Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and brain sciences*, 22(04) :577–660. 27
- Barsalou, L. W. (2009). Simulation, situated conceptualization, and prediction. *Philosophical Transactions of the Royal Society B : Biological Sciences*, 364(1521) :1281–1289. 25
- Barsalou, L. W., Breazeal, C., and Smith, L. B. (2007). Cognition as coordinated non-cognition. *Cognitive Processing*, 8(2) :79–91. 25
- Becchio, C., Sartori, L., Bulgheroni, M., and Castiello, U. (2008). The case of dr. jekyll and mr. hyde : a kinematic study on social intention. *Consciousness and cognition*, 17(3) :557–564. 35
- Becchio, C., Sartori, L., and Castiello, U. (2010). Toward you the social side of actions. *Current Directions in Psychological Science*, 19(3) :183–188. 35
- Beck, B. B. (1980). *Animal tool behavior : the use and manufacture of tools by animals*. New York : Taylor & Francis. 41
- Bellman, R. E. (1957). Dynamic programming. 34
- Berret, B., Darlot, C., Jean, F., Pozzo, T., Papaxanthis, C., and Gauthier, J. P. (2008). The inactivation principle : mathematical solutions minimizing the absolute work and biological implications for the planning of arm movements. *PLoS Comput Biol*, 4(10) :e1000194. 34
- Berthier, N. E., Clifton, R. K., McCall, D. D., and Robin, D. J. (1999). Proximodistal structure of early reaching in human infants. *Experimental brain research*, 127(3) :259–269. 162
- Berthouze, L. and Lungarella, M. (2004). Motor skill acquisition under environmental perturbations : On the necessity of alternate freezing and freeing of degrees of freedom. *Adaptive behavior*, 12(1) :47–64. 162
- Billing, E., Lowe, R., and Sandamirskaya, Y. (2015). Simultaneous planning and action : Neural-dynamic sequencing of elementary behaviours in robot navigation. *Adaptive Behavior*, 9 :1–22. 22, 23, 111
- Billing, E. A. and Sandamirskaya, Y. (2015). Simultaneous planning and action : Neural-dynamic sequencing of elementary behaviors in robot navigation. *Adaptive Behavior*. 21
- Blohm, G. and Crawford, J. (2009). Fields of gain in the brain. *Neuron*, 64 :598–600. 55
- Bonini, L., Serventi, F. U., Simone, L., Rozzi, S., Ferrari, P. F., and Fogassi, L. (2011). Grasping neurons of monkey parietal and premotor cortices encode action goals at distinct levels of abstraction during complex action sequences. *The Journal of Neuroscience*, 31(15) :5876–5886. 22

- Botvinick, M. M. (2008). Hierarchical models of behavior and prefrontal function. *Trends in cognitive sciences*, 12(5) :201–208. 21
- Boucenna, S. (2011). *De la reconnaissance des expressions faciales à une perception visuelle partagée : une architecture sensori-motrice pour amorcer un référencement social d'objets, de lieux ou de comportements*. PhD thesis, Université de Cergy Pontoise. 38
- Braud, R., Pitti, A., and Gaussier, P. (2015). Comparison of absolute and relative strategies to encode sensorimotor transformations in tool-use. In *Development and Learning and on Epigenetic Robotics (ICDL), 2015 IEEE International Conference on*, pages 267–268. IEEE. 29
- Braun, D., Aertsen, A., Wolpert, D., and Mehring, C. (2009). Motor task variation induces structural learning. *Current Biology*, 19 :352–357. 57
- Braun, D., Mehring, C., and Wolpert, D. (2010). Structure learning in action. *Behavioural Brain Research*, 206 :157–165. 57
- Bremner, L. and Andersen, R. (2012). Coding of the reach vector in parietal area 5d. *Neuron*, 75 :342–351. 55
- Brooks, R. A. (1991). Intelligence without representation. *Artificial intelligence*, 47(1-3) :139–159. 10
- Bub, D. N., Masson, M. E., and Cree, G. S. (2008). Evocation of functional and volumetric gestural knowledge by objects and words. *Cognition*, 106(1) :27–58. 24
- Bullock, D., Grossberg, S., and Guenther, F. H. (1993). A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm. *Journal of Cognitive Neuroscience*, 5(4) :408–435. 29, 77, 92
- Burnod, Y., Baraduc, P., Battaglia-Mayer, A., Guigon, E., Koechlin, E., Ferraina, S., Lacquaniti, F., and Caminiti, R. (2000). Parieto-frontal coding of reaching : An integrated framework. 129 :325–46. 30
- Burnod, Y., Grandguillaume, P., Otto, I., Ferraina, S., B Johnson, P., and Caminiti, R. (1992). Visuomotor transformations underlying arm movements toward visual targets : A neural network model of cerebral cortical operations. 12 :1435–53. 30
- Butz, M. V., Sigaud, O., and Gerard, P. (2003). Internal models and anticipations in adaptive learning systems. In *Anticipatory Behavior in Adaptive Learning Systems : Foundations, Theories, and Systems*, pages 86–109. Springer-Verlag. 25, 31
- Butz, M. V., Sigaud, O., Pezzulo, G., and Baldassarre, G. (2006). Anticipations, brains, individual and social behavior : An introduction to anticipatory systems. In *Workshop on Anticipatory Behavior in Adaptive Learning Systems*, pages 1–18. Springer. 31
- Cangelosi, A., Metta, G., Sagerer, G., Nolfi, S., Nehaniv, C., Fischer, K., Tani, J., Belpaeme, T., Sandini, G., Nori, F., et al. (2010). Integration of action and language knowledge : A roadmap for developmental robotics. *Autonomous Mental Development, IEEE Transactions on*, 2(3) :167–195. 16, 46

- Carpenter, G. A. and Grossberg, S. (2010). *Adaptive resonance theory*. Springer. 93
- Chaput, H. H. (2004). *The constructivist learning architecture : A model of cognitive development for robust autonomous robots*. Computer Science Department, University of Texas at Austin. 113, 162
- Chersi, F., Ferrari, P. F., and Fogassi, L. (2011). Neuronal chains for actions in the parietal lobe : a computational model. *PloS one*, 6(11) :e27652. 21
- Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., and Shenoy, K. V. (2012). Neural population dynamics during reaching. *Nature*, 487(7405) :51–56. 65
- Cisek, P. (2007). Cortical mechanisms of action selection : the affordance competition hypothesis. *Philosophical Transactions of the Royal Society B : Biological Sciences*, 362(1485) :1585–1599. 25, 27, 101
- Cisek, P. and Kalaska, J. F. (2010). Neural mechanisms for interacting with a world full of action choices. *Annual review of neuroscience*, 33 :269–298. 24, 25
- Cooper, R. and Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive neuropsychology*, 17(4) :297–338. 21, 22
- Craighero, L., Metta, G., Sandini, G., and Fadiga, L. (2007). The mirror-neurons system : data and models. *Progress in brain research*, 164 :39–59. 23
- Creem-Regehr, S. H. and Lee, J. N. (2005). Neural representations of graspable objects : are tools special ? *Cognitive Brain Research*, 22(3) :457–469. 24
- Cuperlier, N., Quoy, M., and Gaussier, P. (2007). Neurobiologically inspired mobile robot navigation and planning. *Frontiers in neurorobotics*, 1 :3. 40
- Daum, M. M., Zmyj, N., and Aschersleben, G. (2008). Early ontogeny of action perception and production. In *Enacting intersubjectivity : A cognitive and social perspective to the study of interactions*, pages 175–186. IOS Press. 21
- Dayan, P. and Sejnowski, T. J. (1994). Td (λ) converges with probability 1. *Machine Learning*, 14(3) :295–301. 37
- De Rengervé, A., Andry, P., and Gaussier, P. (2015). Online learning and control of attraction basins for the development of sensorimotor control strategies. *Biological cybernetics*, 109(2) :255–274. 29, 40, 49, 82
- de Rengervé, A., Boucenna, S., Andry, P., and Gaussier, P. (2010). Emergent imitative behavior on a robotic arm based on visuo-motor associative memories. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1754–1759. IEEE. 40
- De Rengervé, A., Braud, R., Andry, P., and Gaussier, P. (2012). Behavior adaptation from negative social feedback based on goal awareness. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–6. IEEE. 41, 48

- De Rengervé, A., Hirel, J., Andry, P., Quoy, M., and Gaussier, P. (2011). On-line learning and planning in a pick-and-place task demonstrated through body manipulation. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–6. IEEE. 41
- Decety, J. and Grèzes, J. (2006). The power of simulation : imagining one’s own and other’s behavior. *Brain research*, 1079(1) :4–14. 27
- Dehaene, S. and Changeux, J.-P. (1997). A hierarchical neuronal network for planning behavior. *Proceedings of the National Academy of Sciences*, 94(24) :13293–13298. 21
- Deneve, S. and Pouget, A. (2004). Bayesian multisensory integration and cross-modal spatial links. *J. of Phys.–Paris*, 98 :249–258. 57
- Dennett, D. C. (1978). Current issues in the philosophy of mind. *American Philosophical Quarterly*, pages 249–261. 45
- D’Souza, A., Vijayakumar, S., and Schaal, S. (2001). Learning inverse kinematics. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 298–303. IEEE. 30
- Ellis, R. and Tucker, M. (2000). Micro-affordance : The potentiation of components of action by seen objects. *British journal of psychology*, 91(4) :451–471. 25
- Elsner, B. and Hommel, B. (2001). Effect anticipation and action control. *Journal of experimental psychology : human perception and performance*, 27(1) :229. 43
- Esseily, R., Rat-Fischer, L., O’Regan, K., and Fagard, J. (2013). Understanding the experimenter’s intention improves 16-month-olds’ observational learning of the use of a novel tool. *Cognitive Development*, 28(1) :1–9. 16
- Ferri, F., Campione, G. C., Dalla Volta, R., Gianelli, C., and Gentilucci, M. (2011). Social requests and social affordances : how they affect the kinematics of motor sequences during interactions between conspecifics. *PLoS One*, 6(1) :e15855. 35
- Fitzpatrick, P., Metta, G., Natale, L., Rao, S., and Sandini, G. (2003). Learning about objects through action-initial steps towards artificial cognition. In *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*, volume 3, pages 3140–3145. IEEE. 42
- Flanagan, J. R., Vetter, P., Johansson, R. S., and Wolpert, D. M. (2003). Prediction precedes control in motor learning. *Current Biology*, 13(2) :146–150. 97
- Flash, T. and Hogan, N. (1985). The coordination of arm movements : an experimentally confirmed mathematical model. *The journal of Neuroscience*, 5(7) :1688–1703. 34
- Fogassi, L., Ferrari, P. F., Gesierich, B., Rozzi, S., Chersi, F., and Rizzolatti, G. (2005). Parietal lobe : from action organization to intention understanding. *Science*, 308(5722) :662–667. 24

- Forestier, S. and Oudeyer, P.-Y. (2016). Curiosity-driven development of tool use precursors : a computational model. In *38th Annual Conference of the Cognitive Science Society (CogSci 2016)*, pages 1859–1864. 43
- Francisco, V., Thompson, E., and Rosch, E. (1991). *The Embodied Mind : Cognitive Science and Human Experience*. MIT Press. 10
- Gaussier, P., Joulain, C., Banquet, J.-P., Leprêtre, S., and Revel, A. (2000). The visual homing problem : an example of robotics/biology cross fertilization. *Robotics and autonomous systems*, 30(1) :155–180. 38, 39
- Gaussier, P., Joulain, C., Zrehen, S., Banquet, J.-P., and Revel, A. (1997). Visual navigation in an open environment without map. In *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 2, pages 545–550. IEEE. 69
- Gaussier, P., Moga, S., Quoy, M., and Banquet, J.-P. (1998). From perception-action loops to imitation processes : A bottom-up approach of learning by imitation. *Applied Artificial Intelligence*, 12(7-8) :701–727. 67
- Gaussier, P., Revel, A., Banquet, J.-P., and Babeau, V. (2002). From view cells and place cells to cognitive map learning : processing stages of the hippocampal system. *Biological cybernetics*, 86(1) :15–28. 40
- Gaussier, P. and Zrehen, S. (1995). Perac : A neural architecture to control artificial animals. *Robotics and Autonomous Systems*, 16(2) :291–320. 29, 37, 38, 82
- Gauthier, J.-P., Berret, B., and Jean, F. (2010). A biomechanical inactivation principle. *Proceedings of the Steklov Institute of Mathematics*, 268(1) :93–116. 34
- Georgopoulos, A. P., Schwartz, A. B., and Kettner, R. E. (1986). Neuronal population coding of movement direction. *Science*, 233(4771) :1416–1419. 66
- Gibson, J. (1979). *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, Inc, Mahwah, NJ. 24
- Giovannangeli, C. and Gaussier, P. (2008). Autonomous vision-based navigation : Goal-oriented action planning by transient states prediction, cognitive map building, and sensory-motor learning. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 676–683. IEEE. 38
- Goldenberg, G. and Iriki, A. (2007). From sticks to coffee-maker : mastery of tools and technology by human and non-human primates. *Cortex*, 43 :285–288. 55
- Gopnik, A., Glymour, C., Sobel, D. M., Schulz, L. E., Kushnir, T., and Danks, D. (2004). A theory of causal learning in children : causal maps and bayes nets. *Psychological review*, 111(1) :3. 119
- Gopnik, A., Meltzoff, A. N., and Kuhl, P. K. (2000). *The scientist in the crib : What early learning tells us about the mind*. William Morrow Paperbacks. 119

- Grossberg, S. (1987). The adaptive self-organization of serial order in behavior : Speech, language, and motor control. *Advances in Psychology*, 43 :313–400. 21
- Grush, R. (2004). The emulation theory of representation : motor control, imagery, and perception. *Behavioral and brain sciences*, 27(03) :377–396. 27
- Guerin, F. (2011). Learning like a baby : a survey of artificial intelligence approaches. *The Knowledge Engineering Review*, 26(02) :209–236. 45
- Guerin, F., Kruger, N., and Kraft, D. (2013). A survey of the ontogeny of tool use : from sensorimotor experience to planning. *Autonomous Mental Development, IEEE Transactions on*, 5(1) :18–45. 16, 43
- Guigon, E., Baraduc, P., and Desmurget, M. (2008). Computational motor control : feedback and accuracy. *European Journal of Neuroscience*, 27(4) :1003–1016. 35
- Harnad, S. (1990). The symbol grounding problem. *Physica D : Nonlinear Phenomena*, 42(1-3) :335–346. 9, 38
- Harris, C. M. and Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature*, 394(6695) :780–784. 35
- Hart, S. and Grupen, R. (2011). Learning generalizable control programs. *Autonomous Mental Development, IEEE Transactions on*, 3(3) :216–231. 111
- Hasson, C. and Gaussier, P. (2010). Frustration as a general regulatory mechanism for motivated navigation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4704–4709. IEEE. 39
- Hatze, H. and Buys, J. (1977). Energy-optimal controls in the mammalian neuromuscular system. *Biological cybernetics*, 27(1) :9–20. 34
- Hauf, P., Aschersleben, G., and Prinz, W. (2007). Baby do–baby see ! : How action production influences action perception in infants. *Cognitive Development*, 22(1) :16–32. 21
- Hecht-Nielsen, R. (1987). Counterpropagation networks. *Applied optics*, 26(23) :4979–4984. 38
- Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends in cognitive sciences*, 6(6) :242–247. 27
- Hirel, J. (2011). *Codage hippocampique par transitions spatio-temporelles pour l'apprentissage autonome de comportements dans des tâches de navigation sensori-motrice et de planification en robotique*. PhD thesis, Université de Cergy Pontoise. 39
- Hirel, J., Gaussier, P., and Quoy, M. (2011). Biologically inspired neural networks for spatio-temporal planning in robotic navigation tasks. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 1627–1632. IEEE. 40

- Hoffmann, H., Chen, Z., Earl, D., Mitchell, D., Salemi, B., and Sinapov, J. (2014). Adaptive robotic tool use under variable grasps. *Robotics and Autonomous Systems*, 62(6) :833–846. 42
- Hoffmann, M., Marques, H. G., Hernandez Arieta, A., Sumioka, H., Lungarella, M., and Pfeifer, R. (2010). Body schema in robotics : a review. *Autonomous Mental Development, IEEE Transactions on*, 2(4) :304–324. 15, 42
- Hogan, N. (1984). An organizing principle for a class of voluntary movements. *The Journal of Neuroscience*, 4(11) :2745–2754. 34
- Hol, K. and Treue, S. (2001). Different populations of neurons contribute to the detection and discrimination of visual motion. *Vision research*, 41(6) :685–689. 66
- Hommel, B., Müsseler, J., Aschersleben, G., and Prinz, W. (2001). The Theory of Event Coding (TEC) : a framework for perception and action planning. *The Behavioral and brain sciences*, 24(5) :849–937. 20, 21
- Horn B. K., S. B. G. (1981). Determining optical flow. *International Society for Optics and Photonics*, 17 :319–331. 66
- Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S. (2013). Dynamical movement primitives : learning attractor models for motor behaviors. *Neural computation*, 25(2) :328–373. 43, 162
- Iriki, A., Tanaka, M., and Iwamura, Y. (1996). Coding of modified body schema during tool use by macaque postcentral neurones. *Neuroreport*, 7(14) :2325–2330. 55, 61
- Jacob, P. and Jeannerod, M. (2005). The motor theory of social cognition : a critique. *Trends in cognitive sciences*, 9(1) :21–25. 16
- James, W. (1890). Principles of psychology. 1. New York : Holt. 20, 96
- Jamone, L., Damas, B., Endo, N., Santos-Victor, J., and Takanishi, A. (2012). Incremental development of multiple tool models for robotic reaching through autonomous exploration. *Paladyn, Journal of Behavioral Robotics*, 3(3) :113–127. 42, 110
- Jauffret, A. (2014). *De l'auto-évaluation aux émotions : approche neuromimétique et bayésienne de l'apprentissage de comportements complexes impliquant des informations multimodales*. PhD thesis, Paris 11. 39
- Johnson, S. H. (2000). Thinking ahead : the case for motor imagery in prospective judgements of prehension. *Cognition*, 74(1) :33–70. 16
- Johnson-Frey, S. H. (2004). The neural bases of complex tool use in humans. *Trends in cognitive sciences*, 8(2) :71–78. 16
- Kakei, S., Hoffman, D. S., and Strick, P. L. (1999). Muscle and movement representations in the primary motor cortex. *Science*, 285(5436) :2136–2139. 66

- Kaplan, F. and Hafner, V. V. (2005). Mapping the space of skills : An approach for comparing embodied sensorimotor organizations. In *Development and Learning, 2005. Proceedings., The 4th International Conference on*, pages 129–134. IEEE. 45
- Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6) :718–727. 31
- Kemp, C. C. and Edsinger, A. (2006). Robot manipulation of human tools : Autonomous detection and control of task relevant features. In *Proc. of the Fifth Intl. Conference on Development and Learning*. 42
- Keysers, C. (2004). Demystifying social cognition : Hebbian perspective. *Trends in Cognitive Sciences*, 8 :501–507. 55
- Kluzik, J., Diedrichsen, J., Shadmehr, R., and Bastian, A. J. (2008). Reach adaptation : what determines whether we learn an internal model of the tool or adapt the model of our arm ? *Journal of neurophysiology*, 100(3) :1455–1464. 15
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43 :59–69. 58
- Krüger, N., Geib, C., Piater, J., Petrick, R., Steedman, M., Wörgötter, F., Ude, A., Asfour, T., Kraft, D., Omrcen, D., et al. (2011). Object–action complexes : Grounded abstractions of sensory–motor processes. *Robotics and Autonomous Systems*, 59(10) :740–757. 113, 114
- Lacquaniti, F., Terzuolo, C., and Viviani, P. (1983). The law relating the kinematic and figural aspects of drawing movements. *Acta psychologica*, 54(1-3) :115–130. 34
- Laflaquière, A. (2013). *Approche sensorimotrice de la perception de l'espace pour la robotique autonome*. PhD thesis, Université Pierre et Marie Curie-Paris VI. 19
- Lashley, K. S. (1951). *The problem of serial order in behavior*. Bobbs-Merrill. 21
- Law, J., Shaw, P., Earland, K., Sheldon, M., and Lee, M. (2014). A psychology based approach for longitudinal development in cognitive robotics. *Frontiers in neurorobotics*, 8. 44, 112, 121
- Lee, M. H. (2011). Intrinsic activity : from motor babbling to play. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2, pages 1–6. IEEE. 118, 161
- Lee, M. H., Meng, Q., and Chao, F. (2007). Staged competence learning in developmental robotics. *Adaptive Behavior*, 15(3) :241–255. 162
- Lepretre, S., Gaussier, P., and Cocquerez, J.-p. (2000). From navigation to active object recognition. 69
- Lewkowicz, D., Delevoeye-Turrell, Y., Bailly, D., Andry, P., and Gaussier, P. (2013). Reading motor intention through mental imagery. *Adaptive Behavior*, page 1059712313501347. 35

- Lockman, J. J. (2000). A perception-action perspective on tool use development. *Child development*, pages 137–144. 45
- Longo, M. R. and Bertenthal, B. I. (2006). Common coding of observation and execution of action in 9-month-old infants. *Infancy*, 10(1) :43–59. 21
- Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics : a survey. *Connection Science*, 15(4) :151–190. 44
- Lungarella, M. and Sporns, O. (2006). Mapping information flow in sensorimotor networks. *PLoS Comput Biol*, 2(10) :e144. 45
- Mahé, S., Braud, R., Gaussier, P., Quoy, M., and Pitti, A. (2015). Exploiting the gain-modulation mechanism in parieto-motor neurons : Application to visuomotor transformations and embodied simulation. *Neural Networks*, 62 :102–111. 55, 62, 64
- Maillard, M., Gapenne, O., Hafemeister, L., and Gaussier, P. (2005). Perception as a dynamical sensori-motor attraction basin. In *European Conference on Artificial Life*, pages 37–46. Springer. 38
- Maravita, A. and Iriki, A. (2004). Tools for the body (schema). *Trends in cognitive sciences*, 8(2) :79–86. 15, 50, 55, 61
- McCarty, M. E., Clifton, R. K., and Collard, R. R. (1999). Problem solving in infancy : the emergence of an action plan. *Developmental psychology*, 35(4) :1091. 16
- McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks : The sequential learning problem. *Psychology of learning and motivation*, 24 :109–165. 125
- McGuire, L. and Sabes, P. (2009). Sensory transformations and the use of multiple reference frames for reach planning. *Nature Neuroscience*, 12(8) :1056–1061. 55
- Meltzoff, A. N. (2007). Infants' causal learning : Intervention, observation, imitation. 119
- Miller G. A., Galanter E., . P. K. H. (1960). Plans and the structure of behavior. New York : Holt, Rinehart, Winston. 21
- Möller, R. and Schenck, W. (2008). Bootstrapping cognition from behavior—a computerized thought experiment. *Cognitive Science*, 32(3) :504–542. 25, 26, 27, 97
- Moravec, H. (1988). *Mind children : The future of robot and human intelligence*. Harvard University Press. 9
- Movshon, J. A., Adelson, E. H., Gizzi, M. S., and Newsome, W. T. (1992). The analysis of moving visual patterns. In *MIT Press*. 66
- Mussa-Ivaldi, F. A. and Hogan, N. (1991). Integrable solutions of kinematic redundancy via impedance control. *The International Journal of Robotics Research*, 10(5) :481–491. 30
- Nabeshima, C., Kuniyoshi, Y., and Lungarella, M. (2006). Adaptive body schema for robotic tool-use. *Advanced Robotics*, 20(10) :1105–1126. 42, 110

- Nagai, Y. and Asada, M. (2015). Predictive learning of sensorimotor information as a key for cognitive development. In *Proc. of the IROS 2015 Workshop on Sensorimotor Contingencies for Robotics*. 25
- Nakano, E., Imamizu, H., Osu, R., Uno, Y., Gomi, H., Yoshioka, T., and Kawato, M. (1999). Quantitative examinations of internal representations for arm trajectory planning : minimum commanded torque change model. *Journal of Neurophysiology*, 81(5) :2140–2155. 34
- Nelson, W. L. (1983). Physical principles for economies of skilled movements. *Biological cybernetics*, 46(2) :135–147. 34
- Ngo, H., Luciw, M., Förster, A., and Schmidhuber, J. (2013). Confidence-based progress-driven self-generated goals for skill acquisition in developmental robots. *Frontiers in psychology*, 4. 118
- Nicolescu, M. N. and Matarić, M. J. (2002). A hierarchical architecture for behavior-based robots. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems : part 1*, pages 227–233. ACM. 21
- Nishii, J. and Tani, Y. (2009). Evaluation of trajectory planning models for arm-reaching movements based on energy cost. *Neural computation*, 21(9) :2634–2647. 34
- Norman, J. (2002). Two visual systems and two theories of perception : An attempt to reconcile the constructivist and ecological approaches. *Behavioral and brain sciences*, 25(01) :73–96. 25
- Ogino, M., Hikita, M., Fuke, S., and Asada, M. (2012). Generation of condition-dependent reaching movements based on layered associative networks. In *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*, pages 1–6. IEEE. 21
- Okanoya, K., Tokimoto, N., Kumazawa, N., Hihara, S., and Iriki, A. (2008). Tool-use training in a species of rodent : The emergence of an optimal motor strategy and functional understanding. *PLoS ONE*, 3(3) :e1860. 55, 61
- O’Keefe, J. and Nadel, L. (1979). Precipitous o’keefe & nadel’s the hippocampus as a cognitive map. *Behavioral and Brain Sciences*, 2(04) :487–494. 40
- Olsson, L. A., Nehaniv, C. L., and Polani, D. (2006). From unknown sensors and actuators to actions grounded in sensorimotor perceptions. *Connection Science*, 18(2) :121–144. 45, 76
- O’Regan, J. K. (2011). *Why red doesn’t sound like a bell : Understanding the feel of consciousness*. Oxford University Press. 18
- O’Regan, J. K. and Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and brain sciences*, 24(05) :939–973. 10, 17, 19, 120
- Osiurak, F., Jarry, C., and Le Gall, D. (2010). Grasping the affordances, understanding the reasoning : toward a dialectical theory of human tool use. *Psychological review*, 117(2) :517. 16, 115

- Oudeyer, P.-Y., Kaplan, F., and Hafner, V. V. (2007). Intrinsic motivation systems for autonomous mental development. *Evolutionary Computation, IEEE Transactions on*, 11(2) :265–286. 44, 46, 118, 161
- O’Regan, J. K. and Block, N. (2012). Discussion of j. kevin o’regan’s “why red doesn’t sound like a bell : Understanding the feel of consciousness”. *Review of Philosophy and Psychology*, 3(1) :89–108. 16, 18, 19, 20
- Pellegrini, A. D. (2009). *The role of play in human development*. Oxford University Press, USA. 118
- Perotto, F. S. (2013). A computational constructivist model as an anticipatory learning mechanism for coupled agent–environment systems. *Constructivist Foundations*, 9(1) :46–56. 112
- Pezzulo, G., Candidi, M., Dindo, H., and Barca, L. (2013a). Action simulation in the human brain : twelve questions. *New Ideas in Psychology*, 31(3) :270–290. 27
- Pezzulo, G. and Castelfranchi, C. (2009). Thinking as the control of imagination : a conceptual framework for goal-directed systems. *Psychological Research PRPF*, 73(4) :559–577. 27, 46, 96, 98, 102
- Pezzulo, G., Rigoli, F., and Chersi, F. (2013b). The mixed instrumental controller : using value of information to combine habitual choice and mental simulation. *Frontiers in psychology*, 4 :92. 28
- Pezzulo, G., van der Meer, M. A., Lansink, C. S., and Pennartz, C. M. (2014). Internally generated sequences in learning and executing goal-directed behavior. *Trends in cognitive sciences*, 18(12) :647–657. 28
- Philipona, D., O’Regan, J. K., and Nadal, J.-P. (2003). Is there something out there ? inferring space from sensorimotor dependencies. *Neural computation*, 15(9) :2029–2049. 19
- Piaget, J. and Cook, M. (1952). *The origins of intelligence in children*, volume 8. International Universities Press New York. 43
- Pierce, D. and Kuipers, B. J. (1997). Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92(1) :169–227. 45, 76
- Pitti, A., Blanchard, A., Cardinaux, M., and Gaussier, P. (2012). Gain-field modulation mechanism in multimodal networks for spatial perception. *12th IEEE-RAS International Conference on Humanoid Robots Nov.29-Dec.1, 2012. Business Innovation Center Osaka, Japan*, pages 297–302. 56, 58
- Pitti, A., Braud, R., Mahé, S., Quoy, M., and Gaussier, P. (2013a). Neural model for learning-to-learn of novel task sets in the motor domain. *Frontiers in Psychology*, 4(771). 56
- Pitti, A., Braud, R., Mahé, S., Quoy, M., and Gaussier, P. (2013b). Neural model for learning-to-learn of novel task sets in the motor domain. *Frontiers in Psychology*, 4(771). 58

- Pitti, A., Mori, H., Kouzuma, S., and Kuniyoshi, Y. (2009). Contingency perception and agency measure in visuo-motor spiking neural networks. *IEEE Trans. on Aut. Ment. Dev.*, 1(1) :86–97. 56
- Pontryagin, L., Boltyanskiy, V., Gamkrelidze, R., and MISHCHENKO, Y. (1962). The mathematical theory of optimal processes. *Wiley, New York*. 34
- Pouget, A. and Snyder, L. (1997). Spatial transformations in the parietal cortex using basis functions. *J. of Cog. Neuro.*, 3 :1192–1198. 55, 56, 57, 163
- Prince, C., Helder, N., and Hollich, G. (2005). Ongoing emergence : A core concept in epigenetic robotics. 44, 45
- Prinz, W. (1990). A common-coding approach to perception and action. In *Relationships between perception and action : Current approaches*, pages 167–201. Springer. 21
- Rat-Fischer, L., O'Regan, J. K., and Fagard, J. (2012). The emergence of tool use during the second year of life. *Journal of experimental child psychology*, 113(3) :440–446. 16, 93
- Ratcliff, R. (1990). Connectionist models of recognition memory : Constraints imposed by learning and forgetting functions. *Psychological review*, 97(2) :285–308. 125
- Revel, A. and Andry, P. (2009). Emergence of structured interactions : From a theoretical model to pragmatic robotics. *Neural networks*, 22(2) :116–125. 65
- Rizzolatti, G. and Craighero, L. (2004). The mirror-neuron system. *Annu. Rev. Neurosci.*, 27 :169–192. 23, 24
- Rolf, M. and Asada, M. (2015). What are goals ? and if so, how many ? In *Development and Learning and on Epigenetic Robotics, 2015. ICDL 2015. IEEE 5th International Conference on*. IEEE. 24, 45
- Rolf, M., Steil, J. J., and Gienger, M. (2010). Learning flexible full body kinematics for humanoid tool use. In *Emerging Security Technologies (EST), 2010 International Conference on*, pages 171–176. IEEE. 42, 110
- Rolland de Rengerve, A. (2013). *Apprentissage Interactif en Robotique Autonome : vers de nouveaux types d'IHM*. PhD thesis, Cergy-Pontoise. 40, 41, 47, 48, 50, 67
- Salinas, E. and Thier, P. (2000). Gain modulation : A major computational principle of the central nervous system. *Neuron*, 27 :15—21. 57, 60
- Sandamirskaya, Y. and Schöner, G. (2010). Serial order in an acting system : a multidimensional dynamic neural fields implementation. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, pages 251–256. IEEE. 21
- Schaal, S. and Atkeson, C. G. (1998). Constructive incremental learning from only local information. *Neural Computation*, 10(8) :2047–2084. 46, 124

- Schillaci, G., Hafner, V. V., and Lara, B. (2016). Exploration behaviors, body representations, and simulation processes for the development of cognition in artificial agents. *Frontiers in Robotics and AI*, 3 :39. 27
- Schöner, G. (2008). Dynamical systems approaches to cognition. *Cambridge handbook of computational cognitive modeling*, pages 101–126. 22, 111
- Sigaud, O. and Buffet, O. (2008). *Processus décisionnels de Markov en intelligence artificielle (volume 1)*. Lavoisier. 32, 37
- Singh, S., Jaakkola, T., Littman, M. L., and Szepesvári, C. (2000). Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38(3) :287–308. 37
- Smith, L. and Gasser, M. (2005). The development of embodied cognition : Six lessons from babies. *Artificial life*, 11(1-2) :13–29. 45
- Smith, L. B. and Thelen, E. (2003). Development as a dynamic system. *Trends in cognitive sciences*, 7(8) :343–348. 45, 78
- Smith, P. K. (2009). *Children and play : understanding children's worlds*, volume 12. John Wiley & Sons. 118
- Sommerville, J. A. and Woodward, A. L. (2005). Infants' sensitivity to the causal features of means–end support sequences in action and perception. *Infancy*, 8(2) :119–145. 21
- Srinivasa, N., Bhattacharyya, R., Sundareshwara, R., Lee, C., and Grossberg, S. (2012). A bio-inspired kinematic controller for obstacle avoidance during reaching tasks with real robots. *Neural Networks*, 35 :54–69. 77, 92, 116
- Srinivasa, N. and Grossberg, S. (2008). A head–neck–eye system that learns fault-tolerant saccades to 3-d targets using a self-organizing neural model. *Neural Networks*, 21(9) :1380–1391. 77, 93
- Steele, J., Ferrari, P. F., and Fogassi, L. (2012). From action to language : comparative perspectives on primate tool use, gesture and the evolution of human language. *Philosophical Transactions of the Royal Society B : Biological Sciences*, 367(1585) :4–9. 16
- Steels, L. (2004). The autotelic principle. In *Embodied artificial intelligence*, pages 231–242. Springer. 45
- Stout, D. and Chaminade, T. (2007). The evolutionary neuroscience of tool making. *Neuropsychologia*, 45(5) :1091–1100. 16
- Stout, D. and Chaminade, T. (2009). Making tools and making sense : complex, intentional behaviour in human evolution. *Cambridge Archaeological Journal*, 19(01) :85–96. 16
- Stoytchev, A. (2005). Behavior-grounded representation of tool affordances. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3060–3065. IEEE. 43

- Stoytchev, A. (2009). Some basic principles of developmental robotics. *Autonomous Mental Development, IEEE Transactions on*, 1(2) :122–130. 44
- Stronger, D. and Stone, P. (2006). Towards autonomous sensor and actuator model induction on a mobile robot. *Connection Science*, 18(2) :97–119. 45
- Stulp, F. and Oudeyer, P.-Y. (2012). Adaptive exploration through covariance matrix adaptation enables developmental motor learning. *Paladyn*, 3(3) :128–135. 163
- Sutton, R. S. and Barto, A. G. (1998). *Introduction to reinforcement learning*. MIT Press. 36, 87
- Tanaka, H., Krakauer, J. W., and Qian, N. (2006). An optimization principle for determining movement duration. *Journal of neurophysiology*, 95(6) :3875–3886. 35
- Tani, J. (2003). Symbols and dynamics in embodied cognition : Revisiting a robot experiment. In *Anticipatory Behavior in Adaptive Learning Systems*, pages 167–178. Springer. 45
- Tani, Y. and Nishii, J. (2007). Optimality of reaching movements based on energetic cost under the influence of signal-dependent noise. In *International Conference on Neural Information Processing*, pages 1091–1099. Springer. 34
- Thill, S., Caligiore, D., Borghi, A. M., Ziemke, T., and Baldassarre, G. (2013). Theories and computational models of affordance and mirror systems : an integrative review. *Neuroscience & BioBehavioral Reviews*, 37(3) :491–521. 25
- Thorpe, S., Delorme, A., and Van Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural Networks*, 14 :715–725. 58
- Tikhanoff, V., Pattacini, U., Natale, L., and Metta, G. (2013). Exploring affordances and tool use on the icub. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 130–137. IEEE. 43
- Todorov, E. (2004). Optimality principles in sensorimotor control. *Nature neuroscience*, 7(9) :907–915. 34
- Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological review*, 55(4) :189. 40
- Ugur, E., Oztop, E., and Sahin, E. (2011). Goal emulation and planning in perceptual space using learned affordances. *Robotics and Autonomous Systems*, 59(7) :580–595. 43, 112, 113
- Uno, Y., Kawato, M., and Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint arm movement. *Biological cybernetics*, 61(2) :89–101. 34
- Van Rullen, R., Gautrais, J., Delorme, A., and Thorpe, S. (1998). Face processing using one spike per neurone. *BioSystems*, 48 :229–239. 58
- Van Rullen, R. and Thorpe, S. (2002). Surfing a spike wave down the ventral stream. *Vision Research*, 42 :2593–2615. 58

-
- Varela, F. G., Maturana, H. R., and Uribe, R. (1974). Autopoiesis : the organization of living systems, its characterization and a model. *Biosystems*, 5(4) :187–196. 10, 168
- Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4) :279–292. 37
- Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, University of Cambridge England. 37
- Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., and Thelen, E. (2001). Autonomous mental development by robots and animals. *Science*, 291(5504) :599–600. 45
- Whitebread, D. (2010). Play, metacognition and self-regulation. *Play and learning in the early years*, pages 161–176. 118
- Willatts, P. (1999). Development of means–end behavior in young infants : Pulling a support to retrieve a distant object. *Developmental psychology*, 35(3) :651. 16
- Wolpert, D. M., Doya, K., and Kawato, M. (2003). A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society B : Biological Sciences*, 358(1431) :593–602. 31, 64
- Wolpert, D. M. and Ghahramani, Z. (2000). Computational principles of movement neuroscience. *nature neuroscience*, 3 :1212–1217. 31
- Wolpert, D. M. and Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural networks*, 11(7) :1317–1329. 31, 97
- Wolpert, D. M., Miall, R. C., and Kawato, M. (1998). Internal models in the cerebellum. *Trends in cognitive sciences*, 2(9) :338–347. 31, 32, 35