

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 Δ_s^{Pred}) : la nouvelle loi apprendra alors l'erreur ($\Delta_s^{Pred} - \Delta_s^{Real}$), et non la loi courante constatée (Δ_s^{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.

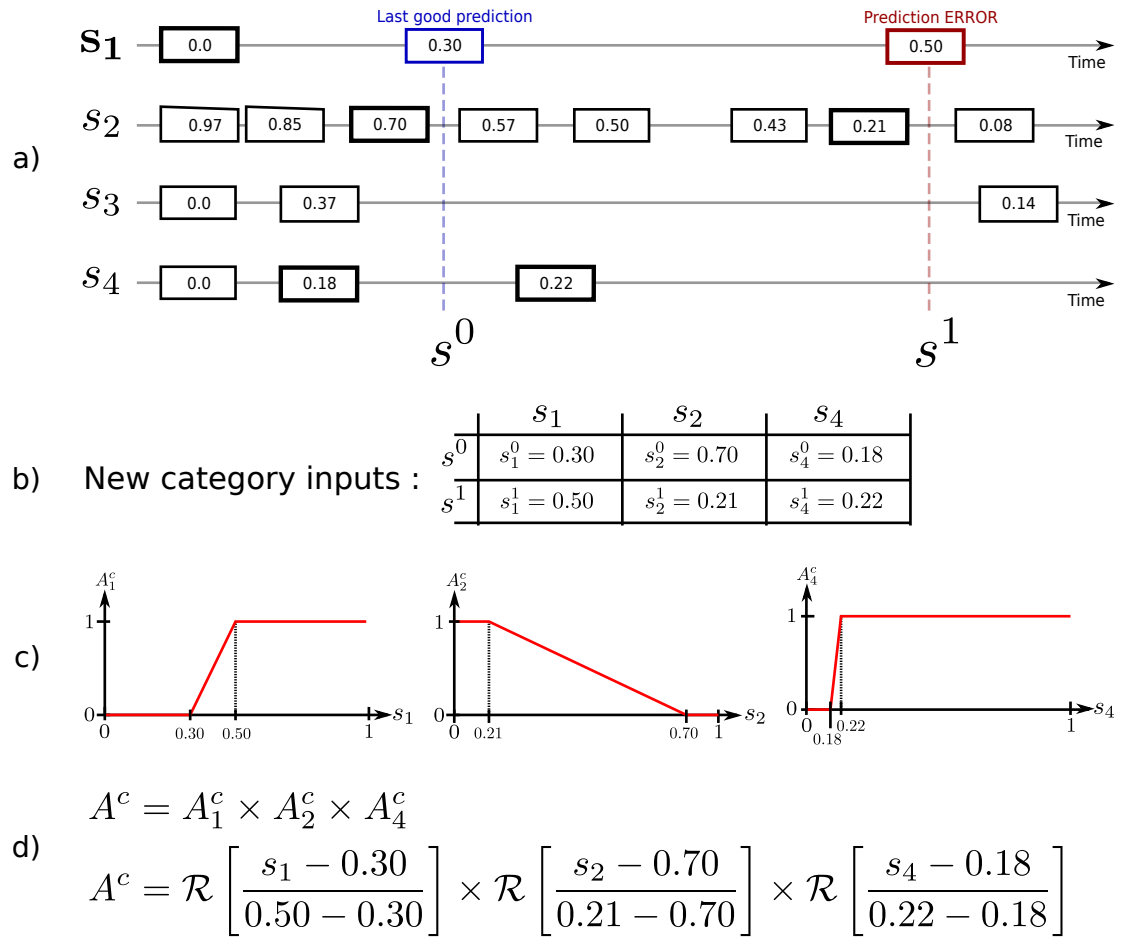


FIGURE 4.1 – Exemple de création de loi d'activation lors du recrutement d'une catégorie, en l'occurrence suite à une erreur constatée sur le senseur s_1 . En a) un exemple de valeurs de 4 senseurs au cours du temps, b) les valeurs des 3 senseurs ayant suffisamment varié depuis la précédente bonne prédiction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

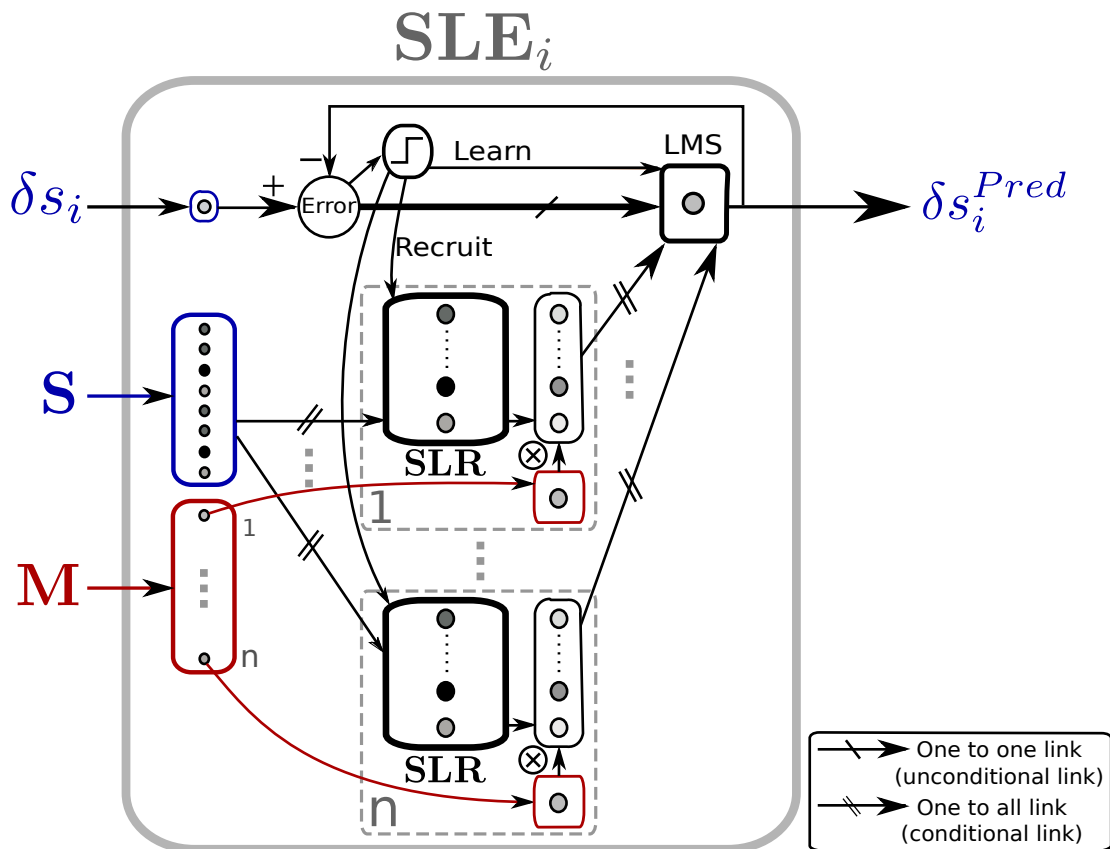


FIGURE 4.2 – Nouvelle version du SLE. Les changements par rapport à la précédente version (fig.3.1) sont en gras : le SLR vient remplacer le SAW et le WTA, et le LMS a désormais l'erreur de prédiction en entrée conditionnelle (et non directement la valeur à prédire).

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  $k^{th}$ 
7:       // neuron in  $SLR^{na}$ , and learning in  $LMS^{na}$ 
8:       for all input sensors  $j$  that have moved since last good prediction, set  $j_1^0, j_1^1, \dots, j_2^0, j_2^1$ 
9:        $w_k^{LMS^{na}} \leftarrow \frac{\Delta s^{Pred} - \Delta s^{Real}}{M_{na}}$ 
10:      end if
11:      // New step : re-initialisation of the predictor
12:       $s_i^{Stored} \leftarrow s_i^{Real}$  ;  $\Delta s_i^{Pred} \leftarrow 0$  ;  $\forall n, M_n \leftarrow 0$ 
13:      end if
14:      // Perform current prediction
15:       $\forall n, M_n \leftarrow M_n + m_n$ 
16:       $\forall n, \left\{ \begin{array}{l} \forall k, A_k^{n,k} \leftarrow \mathcal{R} \left[ \frac{s_{j_1} - s_{j_1}^0}{s_{j_1}^1 - s_{j_1}^0} \right] \times \dots \times \mathcal{R} \left[ \frac{s_{j_2} - s_{j_2}^0}{s_{j_2}^1 - s_{j_2}^0} \right] \\ \forall k, B_k^n \leftarrow m_n A_k^n \end{array} \right.$ 
17:       $\delta s_i^{Pred} \leftarrow \sum_n \sum_k w_{kn}^{LMS^n} B_k^n$ 
18:       $\Delta s_i^{Pred} \leftarrow \Delta s_i^{Pred} + \delta s_i^{Pred}$ 
19:    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).

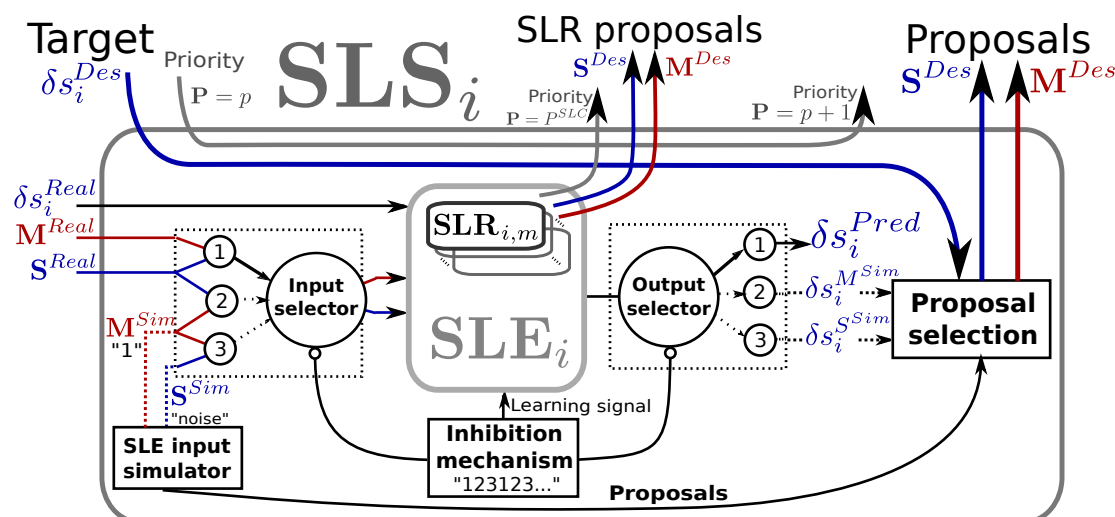


FIGURE 4.3 – Nouvelle version du SLS, avec le SLR. La seule différence avec la précédente version du SLS, fig.3.8, est la présence de nouvelles sorties associées au SLR, mais de même type (S^{Des} , M^{Des} et la priorité P^{SLR} associée).

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-butés 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.

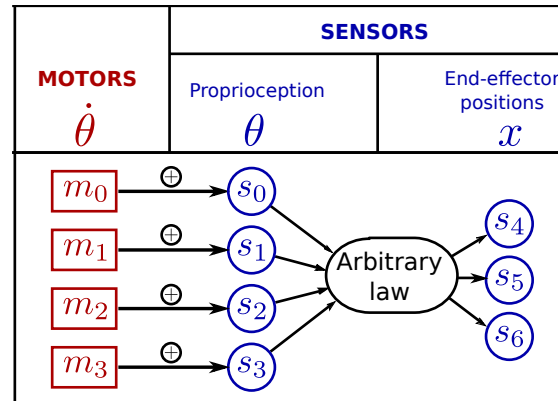


FIGURE 4.4 – Représentation schématique des relations entre les moteurs, la proprioception et la position de l’extrémité du bras. Les notations seront reprises dans les expériences, et une loi sera chaque fois définie.

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

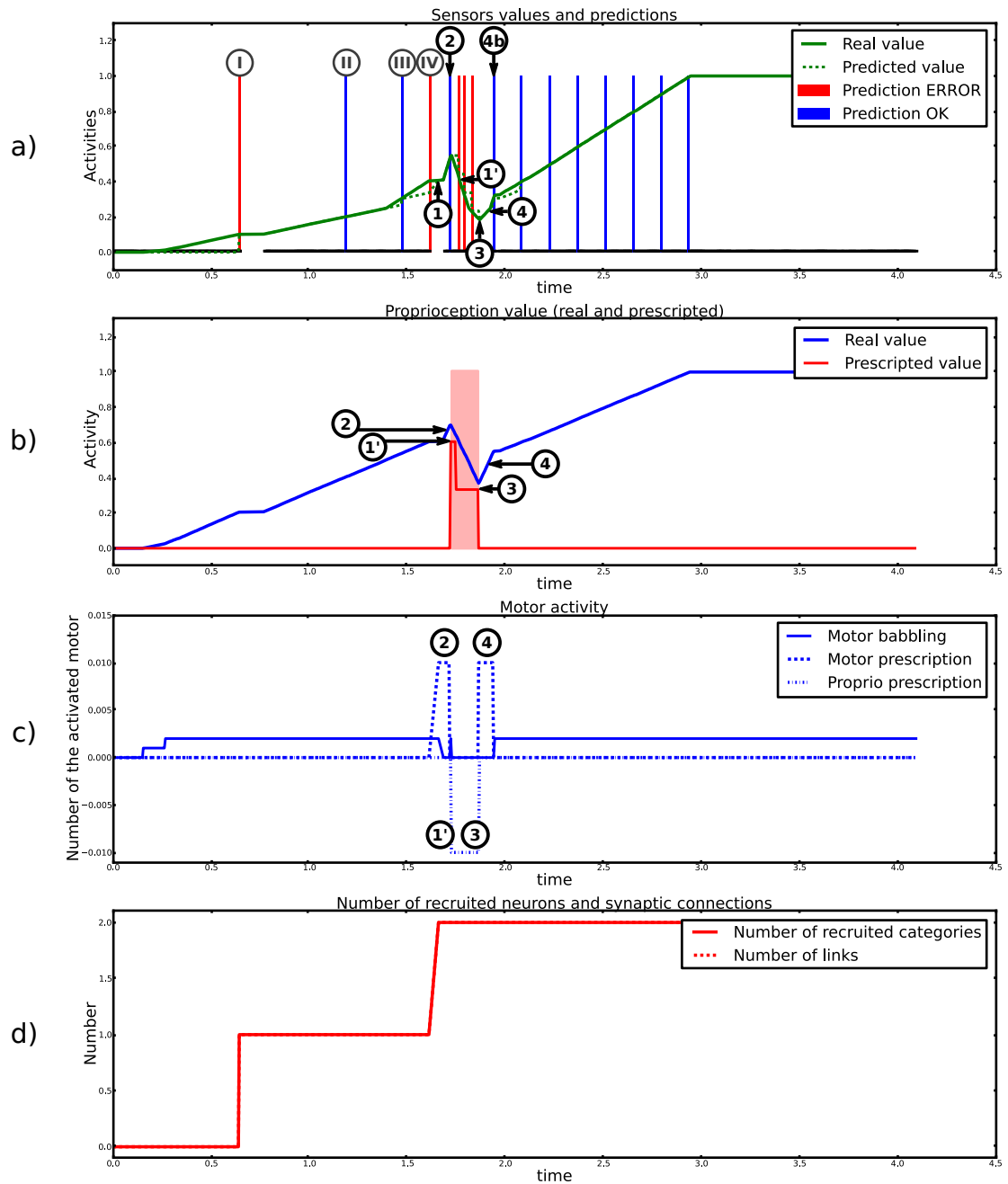


FIGURE 4.5 – Recrutements du SLR avec la loi suivant l’eq.4.10. La courbe a) représente en vert la valeur réelle du capteur s_4 , en pointillé la valeur prédite, et les barres bleues et rouges indiquent respectivement les prédictions correctes ou les erreurs de prédiction. La courbe b) montre en bleu 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 capteur 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 capteur 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 capteurs qui sont des conditions.

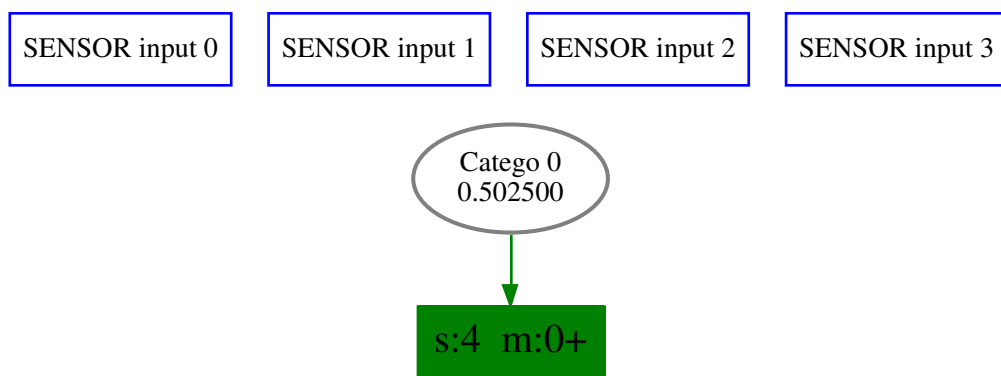


FIGURE 4.6 – Schéma représentant les catégories du SLR du moteur m_0 et du capteur s_4 , pour les mouvements positifs. La valeur associée à la catégorie (0.5025) indique le poids appris par le LMS (voir fig.4.2). Il s'agit ici du recrutement de la première catégorie (et donc du cas particulier, avec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

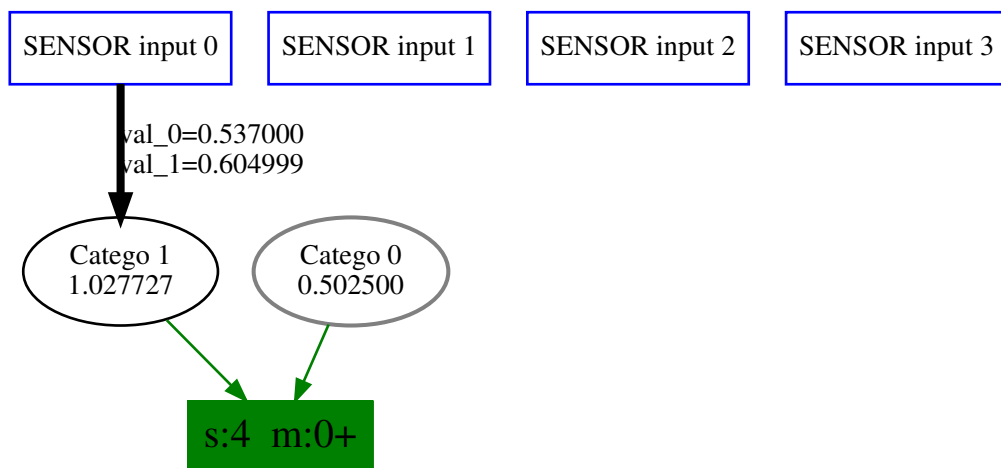


FIGURE 4.7 – Schéma représentant les catégories du SLR du moteur m_0 et du capteur s_4 , pour les mouvements positifs. La valeur associée à la catégorie (1.027727) indique le poids appris par le LMS (voir fig.4.2). Le lien allant de "SENSOR input 0" à "Catego 1" indique que la catégorie recrutée possède un lien vers le capteur 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 $s_0^1 = 0.1$).

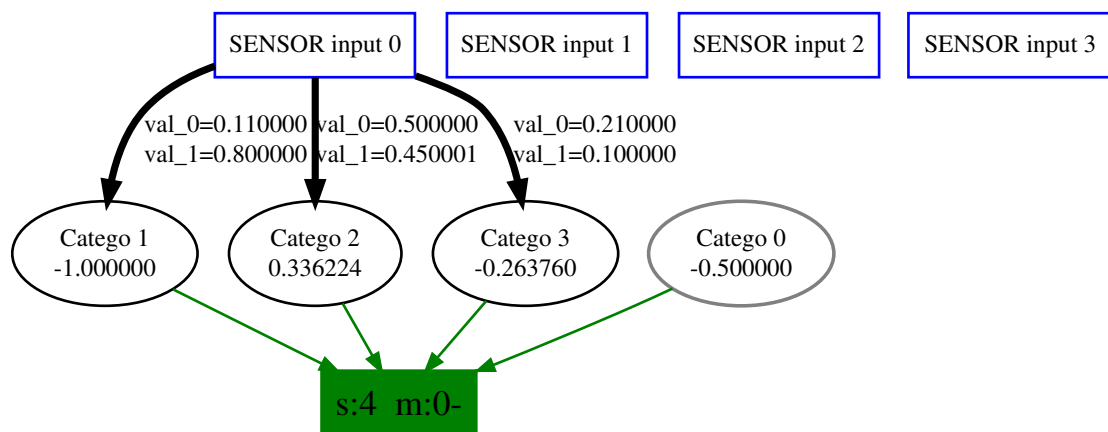


FIGURE 4.8 – Exemple de recrutement non optimal avec un “oubli” temporaire.

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 $s_2^{Stored} = 0$.

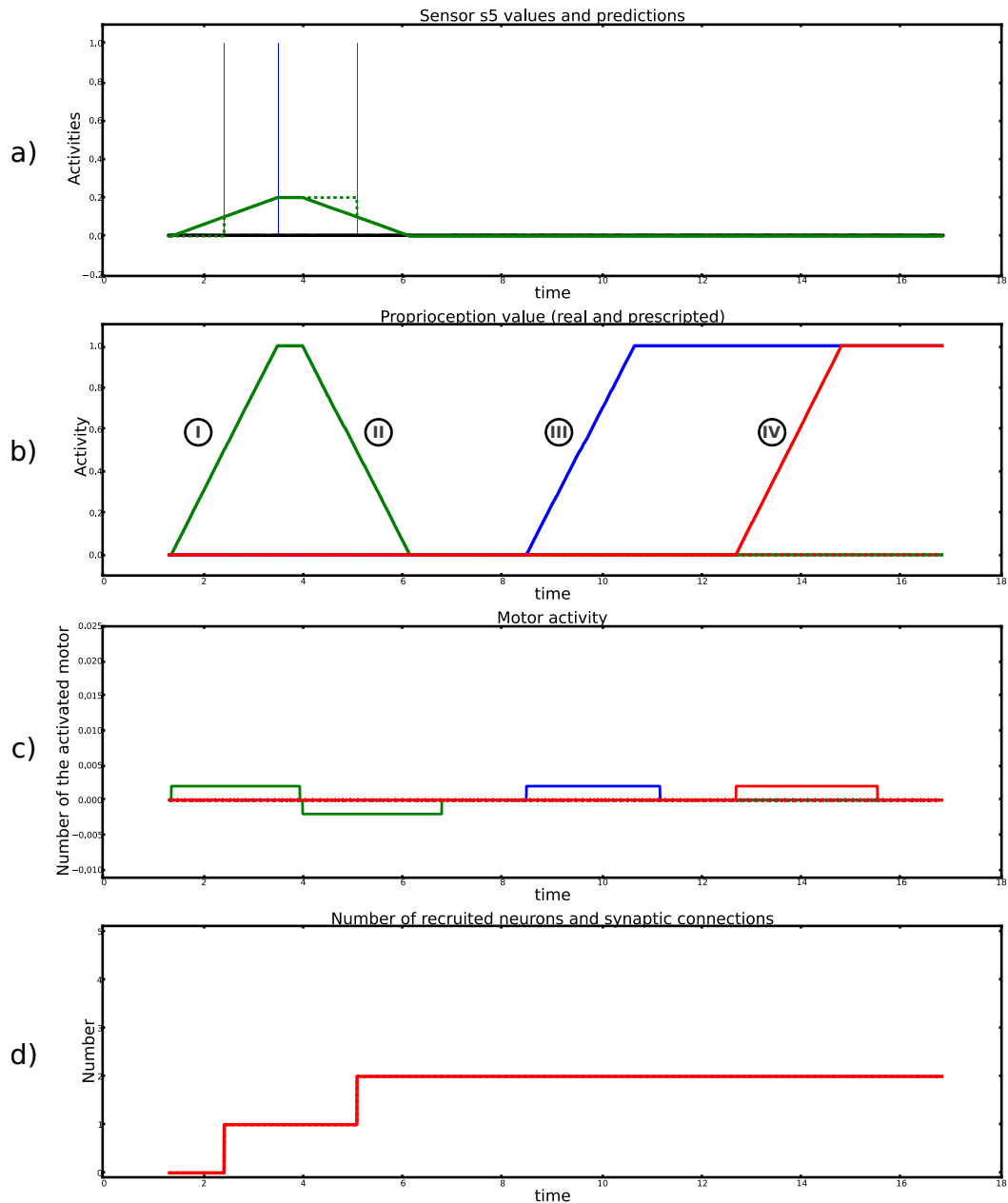


FIGURE 4.9 – Recrutements du SLR_{s_5} avec la loi suivant l'eq.4.9. La courbe a) représente en vert la valeur réelle du capteur s_4 , en pointillé la valeur prédite, et les barres bleues et rouges indiquent les prédictions correctes et les erreurs de prédiction, respectivement. La courbe b) montre les 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.

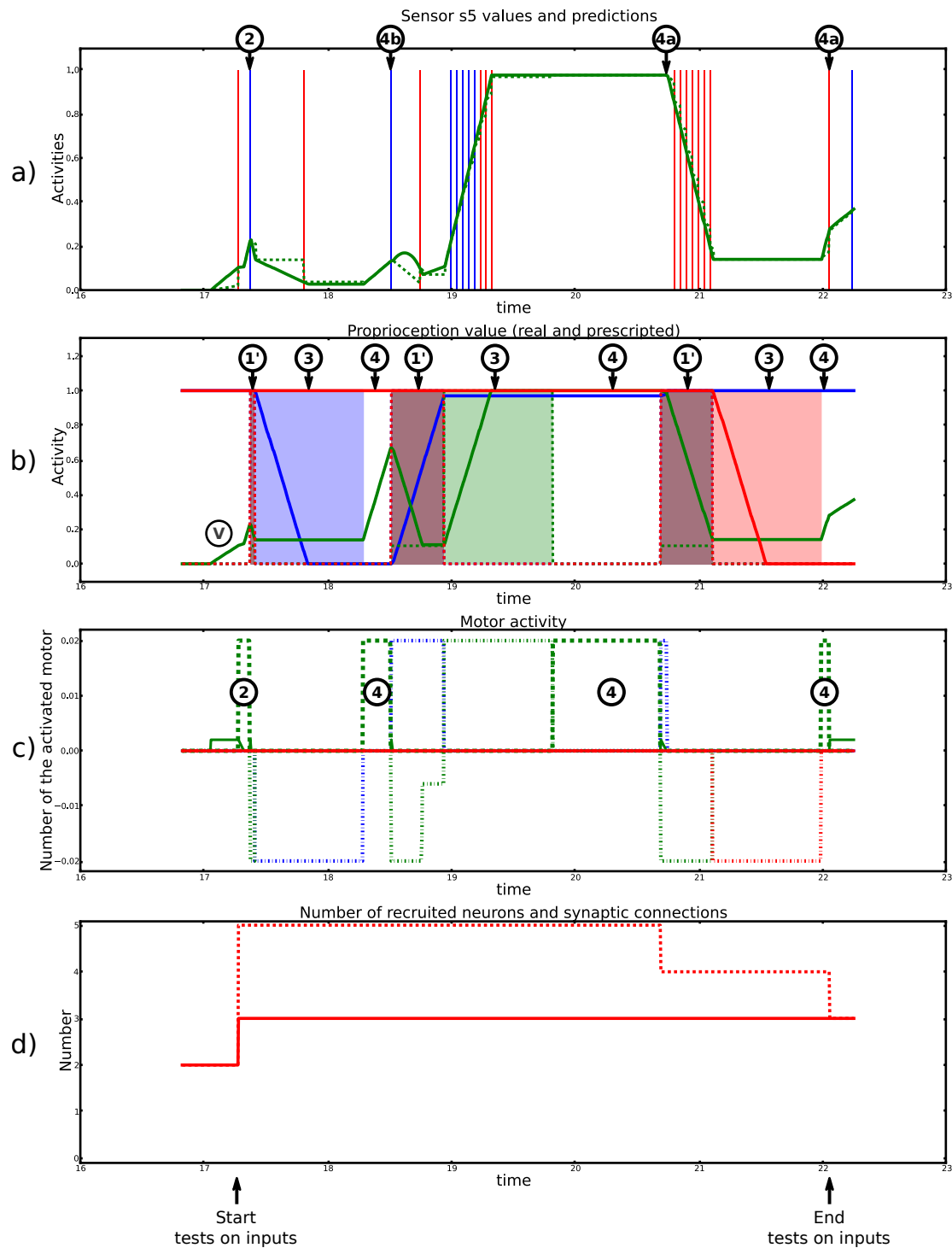


FIGURE 4.10 – Suite de la figure 4.9. On y voit les différentes étapes de recrutement lorsque plusieurs senseurs sont potentiellement pertinents pour catégoriser le nouveau changement de loi observé.

- 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.

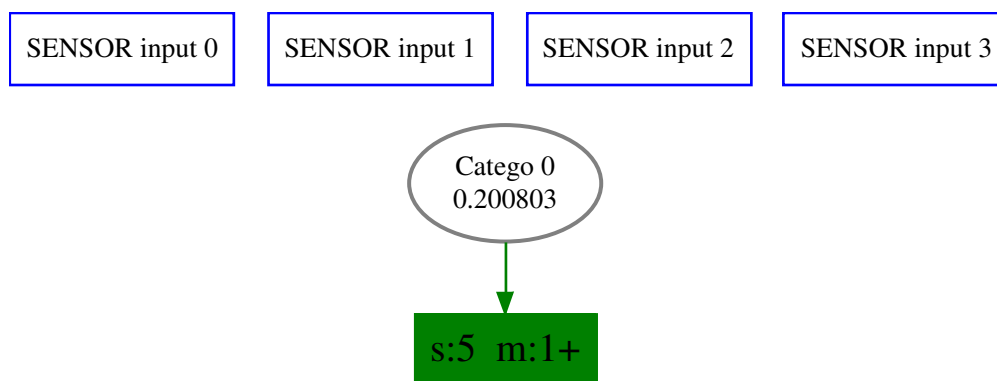


FIGURE 4.11 – Schéma représentant les catégories du SLR du moteur m_1 et du senseur s_4 , pour les mouvements positifs. Recrutement de la première catégorie : la valeur associée à la catégorie 0 (0.200803) indique le poids appris par le LMS (voir fig.4.2).

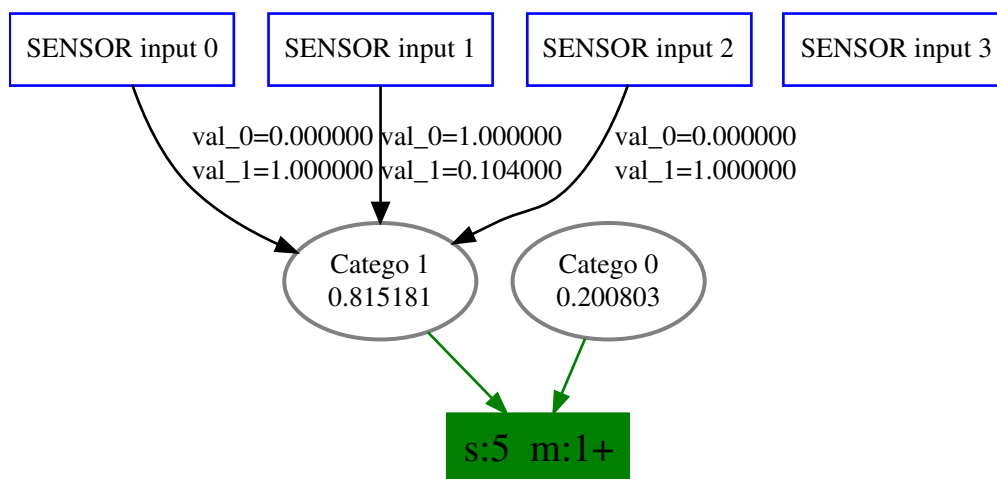


FIGURE 4.12 – Schéma représentant les catégories du SLR du moteur m_1 et du senseur s_4 , pour les mouvements positifs. Réseau au moment du recrutement de la catégorie 2. Les liens allant des “SENSOR input” à “Catego 1” indiquent les liens vers les différents senseurs, et val_0 et val_1 les valeurs s^0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

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

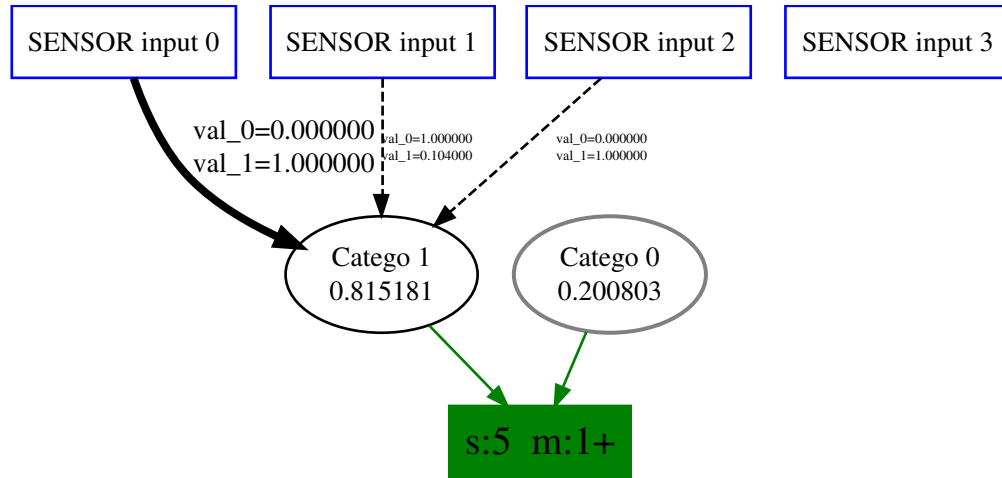


FIGURE 4.13 – Schéma représentant les catégories du SLR du moteur m_1 et du capteur s_4 , pour les mouvements positifs. Réseau à l'issue du recrutement et des tests qui s'en suivent. Les liens d'activations en gras sont conservés, et ceux en pointillés sont désactivés.

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 :

$$x = (L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)) * \cos \theta_0 \quad (4.10)$$

$$y = (L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)) * \sin \theta_0 \quad (4.11)$$

$$z = L_0 + (L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3)) \quad (4.12)$$

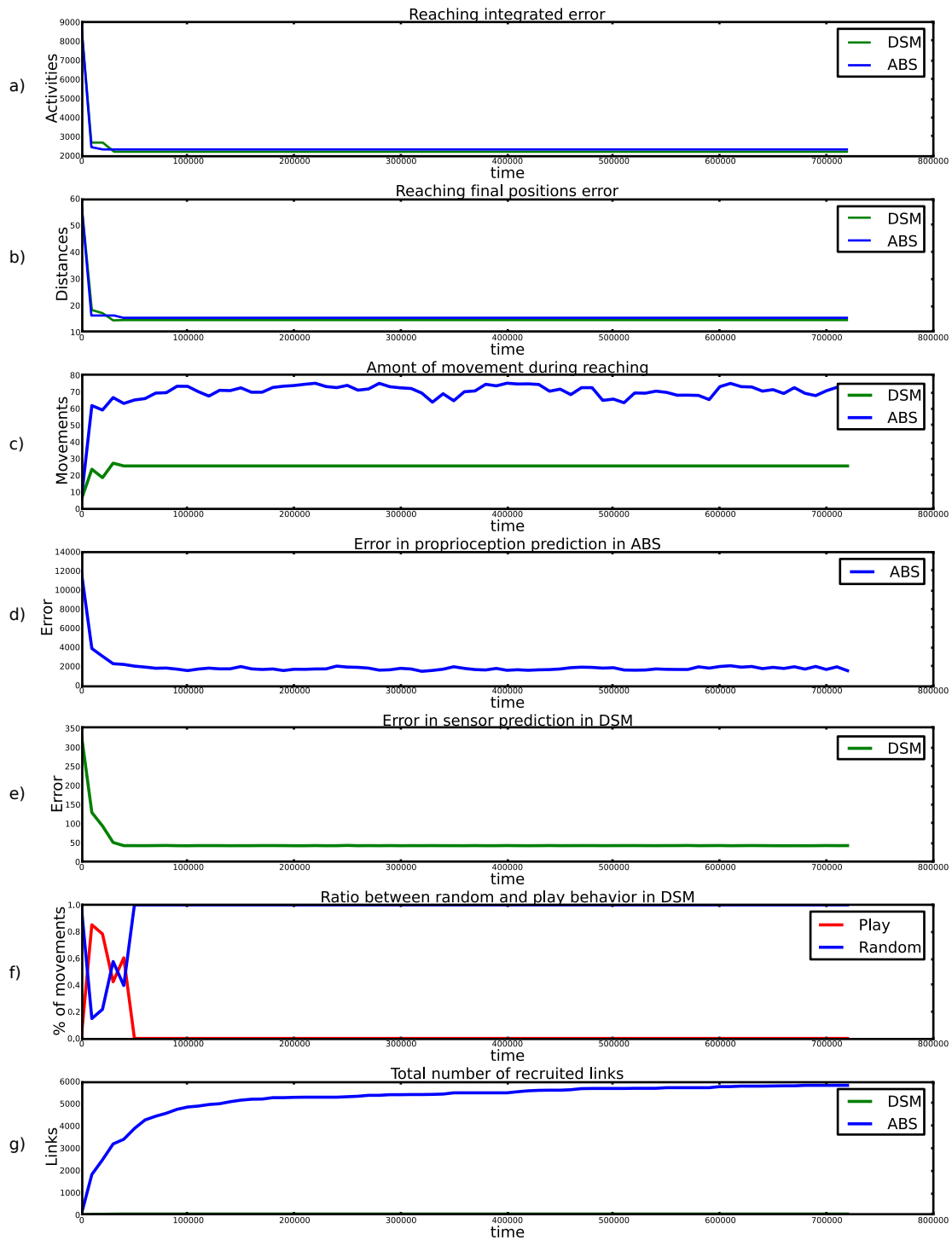


FIGURE 4.14 – Performances comparées entre DSM et le modèle absolu (ABS). a) et b) représentent respectivement l’erreur accumulée, et l’erreur par rapport à la position finale, lors de la tâche d’atteinte. c) montre la quantité de mouvements accumulés pendant les mouvements d’atteinte. d) et e) montrent 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.

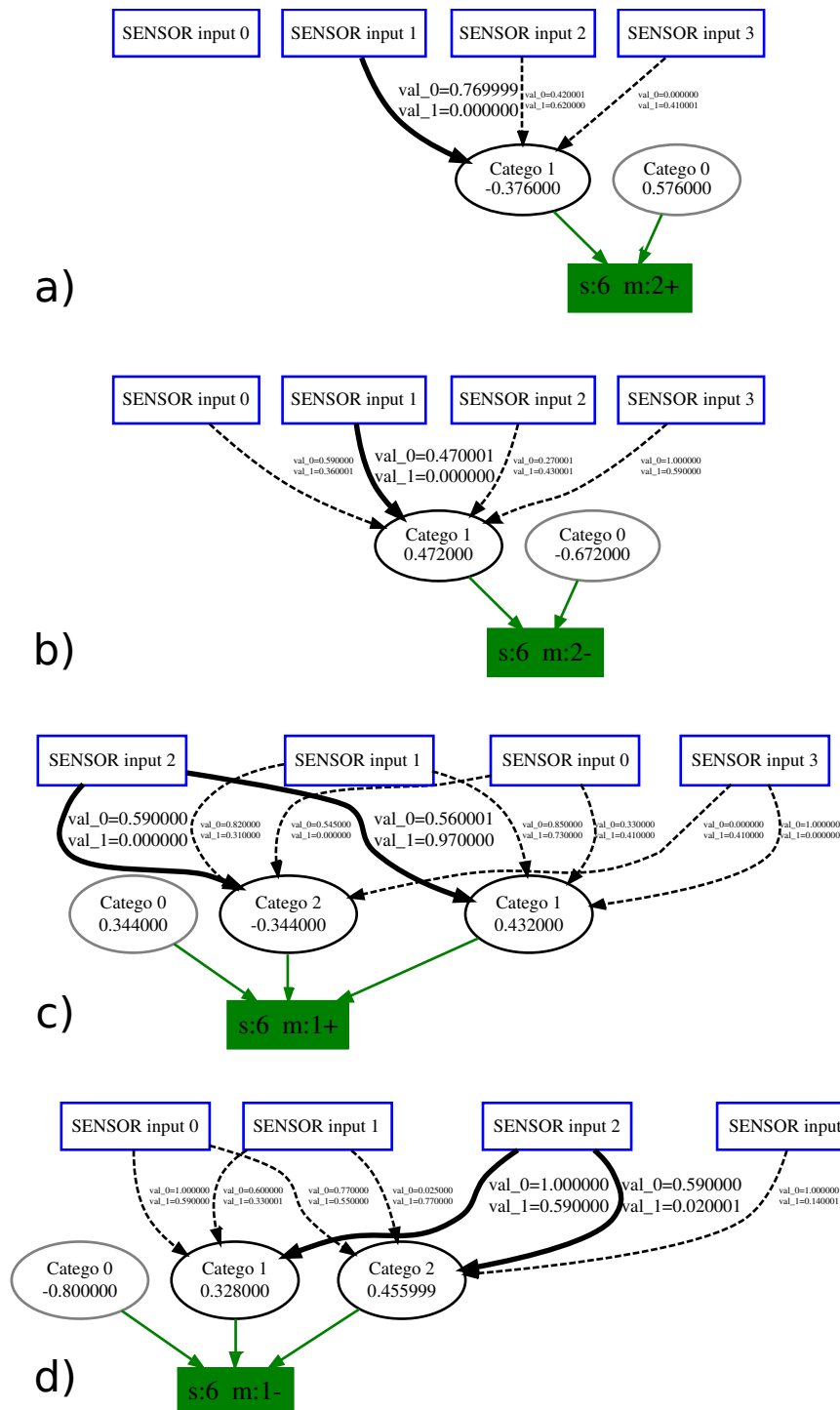


FIGURE 4.15 – Réseaux appris par le SLR concernant le capteur s_6 . Le SLE a découvert deux moteurs pouvant faire varier ce capteur : m_2 , représenté dans a) et b), et m_1 représenté dans c) et d)). Il y a de plus deux réseaux par moteur : ceux pour les grandeurs motrices positives (a) et c)), ainsi 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

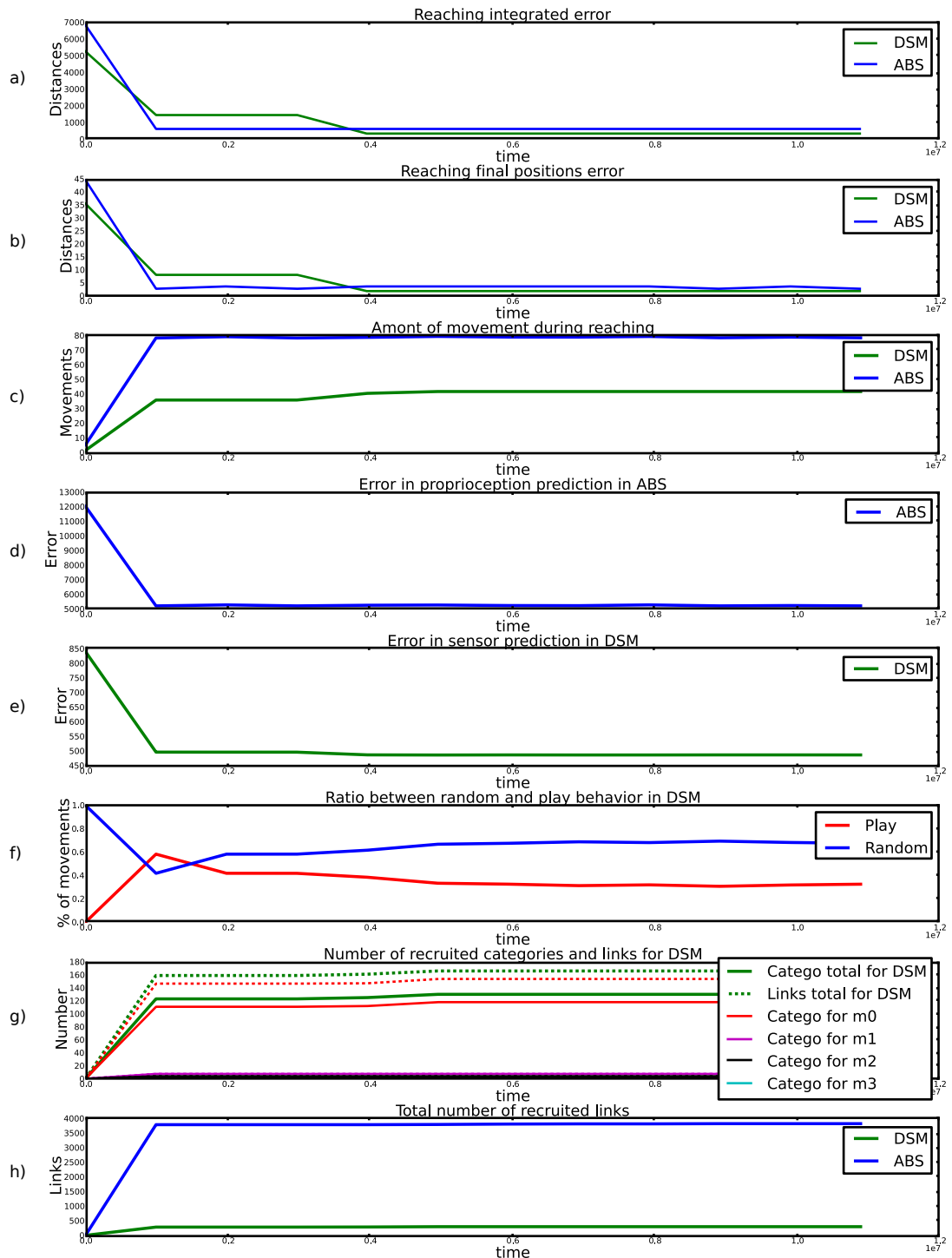


FIGURE 4.16 – Performances comparées entre DSM et le modèle absolu (ABS). a) et b) représentent respectivement l’erreur accumulée, et l’erreur par rapport à la position finale, lors de la tâche d’atteinte. c) montre la quantité de mouvements accumulés pendant les mouvements d’atteinte. d) et e) montre 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.

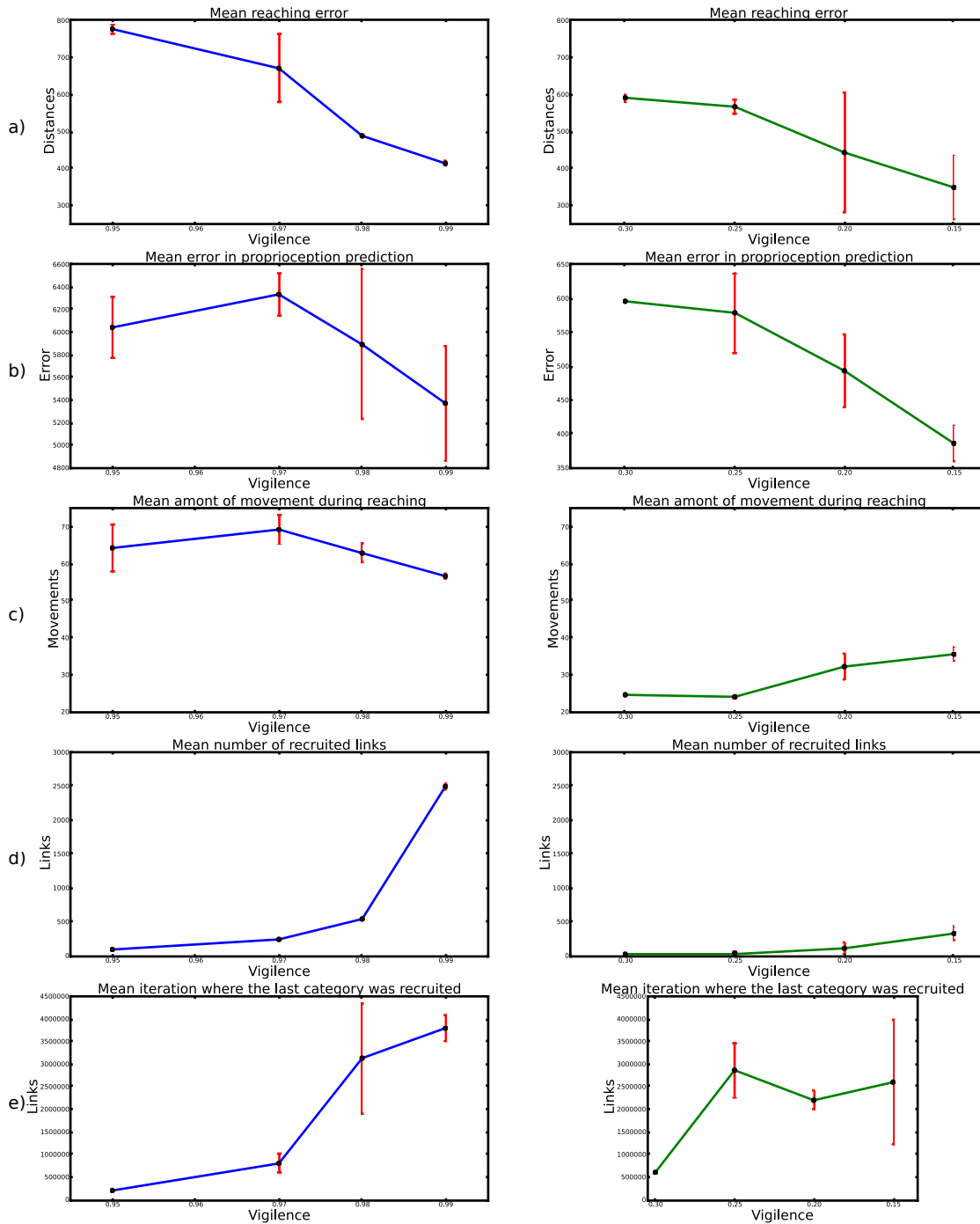


FIGURE 4.17 – Courbes des résultats par vigilance, pour la méthode absolue (colonne de gauche) et pour DSM (colonne de droite), après 5 000 000 d’itérations. a) indique l’erreur moyenne d’atteinte, b) l’erreur moyenne de prédiction, c) la quantité de mouvement, d) le nombre le liens recrutés, et f) l’itération 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.