Apprentissage Interactif en Robotique Autonome : vers de nouveaux types d'IHM

Antoine Rolland de Rengerve

▶ **To cite this version:**

Antoine Rolland de Rengerve. Apprentissage Interactif en Robotique Autonome : vers de nouveaux types d'IHM. Autre. Université de Cergy Pontoise, 2013. Français. NNT : 2013CERG0664 . tel-00969519

HAL Id: tel-00969519

https://theses.hal.science/tel-00969519

Submitted on 27 Nov 2014

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

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




Université de Cergy-Pontoise - Ecole doctorale Sciences et Ingénierie

THÈSE

présentée pour obtenir le titre de DOCTEUR
en Sciences et Techniques de l'Information et de la Communication

---

APPRENTISSAGE INTERACTIF EN ROBOTIQUE AUTONOME : VERS DE NOUVEAUX TYPES D'IHM

---

par

Antoine Rolland de Rengervé

ETIS, UMR8051 ENSEA - Université Cergy-Pontoise - CNRS
6 avenue du Ponceau, 95014 Cergy-Pontoise Cedex, France

P. GAUSSIER,	ETIS, ENSEA/UCP/CNRS	Directeur de thèse
P. ANDRY,	ETIS, ENSEA/UCP/CNRS	Co-encadrant
P. F. DOMINEY,	SBRI, INSERM U846	Rapporteur
P.-Y. OUDEYER,	FLOWERS, INRIA Bordeaux	Rapporteur
R. CHATILA,	ISIR, CNRS/UPMC	Examinateur
F. BEN OUEZDOU,	LISV, UVSQ	Examinateur
J.-C. BAILLIE,	Aldebaran Robotics	Examinateur

Remerciements

Je tiens tout d'abord à remercier Madame Inbar Fijalkow, directrice du laboratoire ETIS, pour m'avoir accueilli au sein du laboratoire ces quatre dernières années.

Je souhaite exprimer toute ma reconnaissance et mon estime aux membres de mon jury de thèse. Je remercie Monsieur Raja Chatila, Directeur de Recherche au CNRS, de m'avoir fait l'honneur de présider ce jury. Je remercie Monsieur Peter Ford Dominey, Directeur de Recherche à l'INSERM, et Monsieur Pierre-Yves Oudeyer, Directeur de Recherche à l'INRIA de Bordeaux (Team Flowers), d'avoir accepté d'être les rapporteurs de ce manuscrit. Enfin, je remercie chacun d'eux, ainsi que Monsieur Fethi Ben Ouezdou, Professeur à l'Université Versailles Saint Quentin, d'être venus assister à ma soutenance et évaluer le travail que j'ai réalisé durant mon doctorat.

Je tiens à remercier mon directeur de thèse, Philippe Gaussier, et mon co-encadrant, Pierre Andry, sans qui cette thèse n'auraient pas eu lieu. Ils ont su me guider et me faire progresser tout en me laissant suffisamment d'autonomie pour que je puisse évoluer et m'épanouir dans mon projet de recherche. Je tiens à souligner la complémentarité de leur encadrement respectif, me permettant d'apprendre l'importance d'être exigeant mais aussi de savoir capitaliser sur ce qui a déjà été réalisé pour aller à l'essentiel. Bien sûr, je les remercie aussi pour les nombreux échanges scientifiques que nous avons eu et pour la formation de chercheur que j'ai ainsi acquise auprès d'eux. Enfin, je tiens à souligner leur présence et leur soutien pour m'aider quand j'en avais besoin, notamment lorsqu'il fallait finir de corriger - presque - en urgence le manuscrit et la présentation pour la thèse.

L'environnement humain au sein du laboratoire joue une part importante dans la qualité du travail réalisé. Ma réflexion scientifique s'est ainsi nourrie des multiples échanges que j'ai eu au sein de l'équipe Neurocybernétique. Je remercie la bande des doctorants de Saint Martin : ceux qui m'ont précédé et sont déjà docteurs (Christophe Giovannangeli, Matthieur Lagarde, Sofiane Boucenna, Cyril Hasson, Julien Hirel) et ceux qui le seront bientôt (Abdelhak Chatty, David Bailly, Adrien Jauffret, Souheil Hanoune, Pierre Delarboulas, Ali Karaazouen, Caroline Grand, Raphael Braud, et Artem Melnyk, que je n'oublie pas même s'il est en Ukraine). J'ai beaucoup aimé travailler et discuter avec eux. Au delà des échanges techniques et scientifiques, je conserverai en mémoire ces nombreux débats passionnés sur la science, la religion, etc., qui n'auraient pas été les mêmes sans Messieurs Arnaud Blanchard et Frederic Demelo, les indispensables ingénieurs de Neurocyber. Je les remercie pour nos collaboration et toute l'aide qu'ils m'ont apportée durant ces années de travail informatique et électronique.

Je pense aussi à tous les enseignants-chercheurs et aux personnels administratifs et informatiques du laboratoire ETIS et je les remercie pour leur aide et leur soutien ces quatre années au laboratoire. Je tiens à remercier tout particulièrement Astrid et Nelly, secrétaires d'ETIS à Saint Martin pour leur aide précieuse et leur bonne humeur contagieuse. Enfin, je souhaite remercier Ghilès Mostafaoui, Alexandre Pitti, Mathias Quoy, Nicolas Cuperlier, Patrick Hénaff, Laurence Haffemeister et Philippe Laroque, enseignants-chercheurs dans l'équipe Neurocybernétique pour nos échanges scientifiques, pour leur aide et pour leurs encouragements.

Je n'oublie pas les autres doctorants, docteurs et stagiaires d'ETIS qui ont influencé directement ou indirectement mon vécu de cette aventure qu'est le doctorat. La liste est trop longue pour que je puisse être exhaustif, ils se reconnaîtront. Je nommerai toutefois Mehdi Badr et Jean-Christophe Sibel qui ont commencé leur doctorat en même temps que moi et dont l'exemple (pour la ténacité face à la difficulté de finir son doctorat) m'a incité à tenir bon jusqu'au bout.

Remerciements

De même pour Romain Tajan qui, bien qu'ayant commencé un an après moi, a soutenu quelques jours avant moi. Je remercie Jean-Christophe de m'avoir soutenu jusqu'au bout en faisant le déplacement depuis Rennes pour assister à ma soutenance.

D'autre part, je dois aussi un grand merci à l'équipe pédagogique du département d'informatique qui m'a fait découvrir l'enseignement et m'a accompagné durant mes 3 années de thèse et cette année supplémentaire d'ATER. Je pense aux secrétaires, présentes et efficaces, Maryse Zindovic, Dominique Courmont, Nadia Beouch et Koulou Chanfi, ainsi qu'aux responsables de cours qui m'on fait confiance pour enseigner aux étudiants : Virginie Sans, Brahim Derdouri, Marc Lemaire, Ghilès Mostafaoui et Benoît Miramont.

D'un point de vue plus personnel, je remercie ma famille pour m'avoir toujours soutenu pendant toutes ces années (pas seulement de doctorat) et pour avoir accepté les concessions faites aux deadlines toujours pressantes. Je suis heureux et fier d'avoir pu présenter ma soutenance de thèse devant mes parents et d'avoir pu partager avec eux cet instant.

Je remercie tous mes amis qui, même si j'ai souvent dû privilégier mon travail de thèse, ont continué de penser à moi et m'ont fait participer à leurs vies de près ou de loin. En particulier, je remercie Damien, meilleur ami de toujours, qui a pris de son temps pour venir assister à ma soutenance. Sa présence m'a touché.

Pour conclure, il me reste une dernière personne à remercier bien que le terme remercier me semble un peu faible pour exprimer tout ce que je dois à Leila, mon Amour et ma Moitié. Avec tout ce qu'elle m'a apporté et m'apporte encore chaque jour, ses encouragements et son soutien inconditionnel ont certainement autant contribué à la réussite de cette thèse que tous les échanges scientifiques. L'aventure du doctorat s'achève pour laisser place à l'aventure du mariage, et je suis confiant car je sais que nous la vivrons ensemble comme la précédente.

Résumé

Un robot autonome collaborant avec des humains doit être capable d'apprendre à se déplacer et à manipuler des objets dans la même tâche. Dans une approche classique, on considère des modules fonctionnels indépendants gérant les différents aspects de la tâche (navigation, contrôle du bras...). A l'opposé, l'objectif de cette thèse est de montrer que l'apprentissage de tâches de natures différentes peut être abordé comme un problème d'apprentissage d'attracteurs sensorimoteurs à partir d'un petit nombre de structures non spécifiques à une tâche donnée. Nous avons donc proposé une architecture qui permet l'apprentissage et l'encodage d'attracteurs pour réaliser aussi bien des tâches de navigation que de contrôle d'un bras.

Comme point de départ, nous nous sommes appuyés sur un modèle inspiré des cellules de lieu pour la navigation d'un robot autonome. Des apprentissages en ligne et interactifs de couples lieu/action sont suffisants pour faire émerger des bassins d'attraction permettant à un robot autonome de suivre une trajectoire. En interagissant avec le robot, on peut corriger ou orienter son comportement. Les corrections successives et leur encodage sensorimoteur permettent de définir le bassin d'attraction de la trajectoire. Ma première contribution a été d'étendre ce principe de construction d'attracteurs sensorimoteurs à un contrôle en impédance pour un bras robotique. Lors du maintien d'une posture proprioceptive, les mouvements du bras peuvent être corrigés par une modification en-ligne des commandes motrices exprimées sous la forme d'activations musculaires. Les attracteurs moteurs résultent alors des associations simples entre l'information proprioceptive du bras et ces commandes motrices. Dans un second temps, j'ai montré que le robot pouvait apprendre des attracteurs visuo-moteurs en combinant les informations proprioceptives et visuelles. Le contrôle visuo-moteur correspond à un homéostat qui essaie de maintenir un équilibre entre ces deux informations. Dans le cas d'une information visuelle ambiguë, le robot peut percevoir un stimulus externe (e.g. la main d'un humain) comme étant sa propre pince. Suivant le principe d'homéostasie, le robot agira pour réduire l'incohérence entre cette information externe et son information proprioceptive. Il exhibera alors un comportement d'imitation immédiate des gestes observés. Ce mécanisme d'homéostasie, complété par une mémoire des séquences observées et l'inhibition des actions durant l'observation, permet au robot de réaliser des imitations différées et d'apprendre par observation. Pour des tâches plus complexes, nous avons aussi montré que l'apprentissage de transitions peut servir de support pour l'apprentissage de séquences de gestes, comme c'était le cas pour l'apprentissage de cartes cognitives en navigation. L'utilisation de contextes motivationnels permet alors le choix entre les différentes séquences apprises.

Nous avons ensuite abordé le problème de l'intégration dans une même architecture de comportements impliquant une navigation visuomotrice et le contrôle d'un bras robotique pour la préhension d'objets. La difficulté est de pouvoir synchroniser les différentes actions afin que le robot agisse de manière cohérente. Les comportements erronés du robot sont détectés grâce à l'évaluation des actions proposées par le modèle vis à vis des corrections imposées par le professeur humain. Ces situations sont apprises sous la forme de contextes multimodaux modulant la sélection d'action afin d'adapter le comportement du robot et qu'il reproduise la tâche désirée.

Pour finir, nous présentons les perspectives de ce travail en terme de contrôle sensorimoteur, pour la navigation comme pour le contrôle d'un bras robotique, et son extension aux questions d'interface Homme/robot. Nous insistons sur le fait que différents types d'imitation peuvent être le fruit des propriétés émergentes d'une architecture de contrôle sensorimotrice.

Mots-clés : contrôle sensorimoteur, apprentissage interactif, réseaux de neurones artificiels, robotique autonome, imitation, systèmes dynamiques

Abstract

An autonomous robot collaborating with humans should be able to learn tasks involving navigation and object manipulation together. In a classical approach, the different aspects of the task (navigation, arm control,...) are managed by independent functional modules. To the contrary, the goal of this thesis is to show that learning tasks of different kinds can be tackled by learning sensorimotor attractors from a few task-nonspecific structures. We thus proposed an architecture which can learn and encode attractors to perform navigation tasks as well as arm control.

We started with a model based on place-cells used in navigation of autonomous robots. On-line and interactive learning of place-action couples can let attraction basins emerge, allowing an autonomous robot to follow a trajectory. The robot behavior can be corrected and guided as a result of interaction. The successive corrections and their sensorimotor coding enable to define the attraction basin of the trajectory. My first contribution was to adapt the principle of sensorimotor attractor building to the impedance control of a robot arm. While a proprioceptive posture is maintained, the arm movements can be corrected by modifying on-line the motor command given by muscular activations. Motor attractors result from the simple associations between these motor commands and the proprioceptive information of the arm. I then showed that our robot could learn visuomotor attractors by associating the combined proprioceptive and visual information with motor attractors. The visuomotor control corresponds to a homeostatic system trying to maintain an equilibrium between these two kinds of information. In the case of ambiguous visual information, the robot may perceive an external stimulus (e.g. a human hand) as its own hand. According to the principle of homeostasis, the robot will act to reduce the incoherence between this external information and its proprioceptive information. It then displays a behavior of imitation of the immediately observed gestures. This homeostasis mechanism, completed by a memory of the observed sequences and the capability to inhibit mouvements during the observation phase, enables a robot to perform deferred imitation and learning by observation. In the case of more complex tasks, we also showed that learning transitions can be a basis for learning sequences of gestures, like in the case of cognitive map learning in navigation. Activated motivational contexts then enables to choose between different learned sequences.

We then addressed the issue of integrating in the same architecture behaviors involving visuomotor navigation and robotic arm control to grab objects. The difficulty is the synchronization of the different actions so the robot can act coherently. Erroneous behaviors of the robot are detected by comparing the actions predicted by the model with the actions forced by the human teacher during behavior corrections. These situations can be encoded as multimodal contexts participating in the action selection process and the adaptation of the robot behavior so that it can reproduce the desired task.

Finally, we will present the perspectives of this work in terms of sensorimotor control, for both navigation and robotic arm control, and its link to human robot interface issues. We will also insist on the fact that different kinds of imitation behavior can result from the emergent properties of a sensorimotor control architecture.

Keywords: sensorimotor control, interactive learning, artificial neural networks, autonomous robotics, imitation, dynamical systems

Table des matières

Ne me dites pas que ce problème est difficile. S'il n'était pas difficile, ce ne serait pas un problème.

– Maréchal Foch

Introduction

Contexte

Dans les premières années de la robotique, le paradigme de l'Intelligence Artificielle Classique s'est focalisé sur les problèmes de manipulation des symboles et du raisonnement. La résolution de ces problèmes a énormément progressé mais sans forcément aider à la réalisation de robots autonomes : la difficulté majeure de la robotique se trouve dans l'interaction avec l'environnement. Il s'agit en particulier de faire le lien entre les représentations manipulées et la réalité (Symbol Grounding Problem [Harnad, 1990]). Une solution partielle est de faire évoluer les robots dans des environnements maîtrisés mais cela implique des contraintes incompatibles avec une robotique accessible au grand public. Les tâches que devrait réaliser un robot ne peuvent être toutes préprogrammées à l'avance. Un robot devrait donc apprendre de nouvelles tâches au fur et à mesure de son existence. Dans le cadre d'une collaboration Homme-robot, ce dernier pourrait apprendre plus rapidement grâce à l'humain. Mais un humain non expert pourra-t-il enseigner à un robot comment réaliser de nouvelles tâches sans le reprogrammer ? Il faudrait alors que l'apprentissage du robot s'appuie sur une interaction qui soit intuitive et naturelle pour l'Homme. Bien que le développement d'une représentation partagée basée sur le langage naturel [Steels and Baillie, 2003] puisse faciliter l'interaction, le langage n'est pas la seule manière de communiquer et d'interagir. Dans un cadre non-verbal, comment un robot autonome peut-il apprendre de nouvelles tâches en situation d'interaction ?

Dans cette thèse, nous nous intéresserons à l'apprentissage par imitation ou démonstration de tâches quotidiennement accomplies par les humains : ranger une pièce. Considérons un robot devant réaliser ce genre de tâche dans un environnement humain (Figure 1). Durant la tâche, le robot explore une pièce. Lorsqu'il repère un objet à ranger, il se déplacera vers lui. Une fois arrivé à proximité de l'objet, le robot doit l'attraper en utilisant son bras. Le robot observera l'objet de plus près, voire le soupèsera ou le secouera. Par exemple, il pourra s'assurer qu'une boite est vide à partir de son poids ou du son qu'elle produit quand elle est secouée. En fonction du résultat, le robot reposera alors l'objet sur place ou l'emmènera dans un lieu approprié (dans un placard, à la poubelle…). Une fois arrivé, il accomplira alors la série de gestes nécessaires pour poser l'objet sur l'étagère dans le placard ou bien pour laisser tomber l'objet dans la poubelle. Enfin, le robot pourra reprendre son parcours à la recherche d'autres objets à ranger. Cette tâche quotidiennement réalisée par les humains se révèle complexe à reproduire et à apprendre pour un robot. Tout d'abord cette tâche implique des séquences d'actions multiples qui doivent être exécutées suivant un ordre précis et selon les situations particulières pour obtenir le résultat attendu. Chacune des actions comme naviguer entre les différents endroits ou attraper des objets va impliquer le traitement de multiples informations provenant de différentes modalités sensorielles

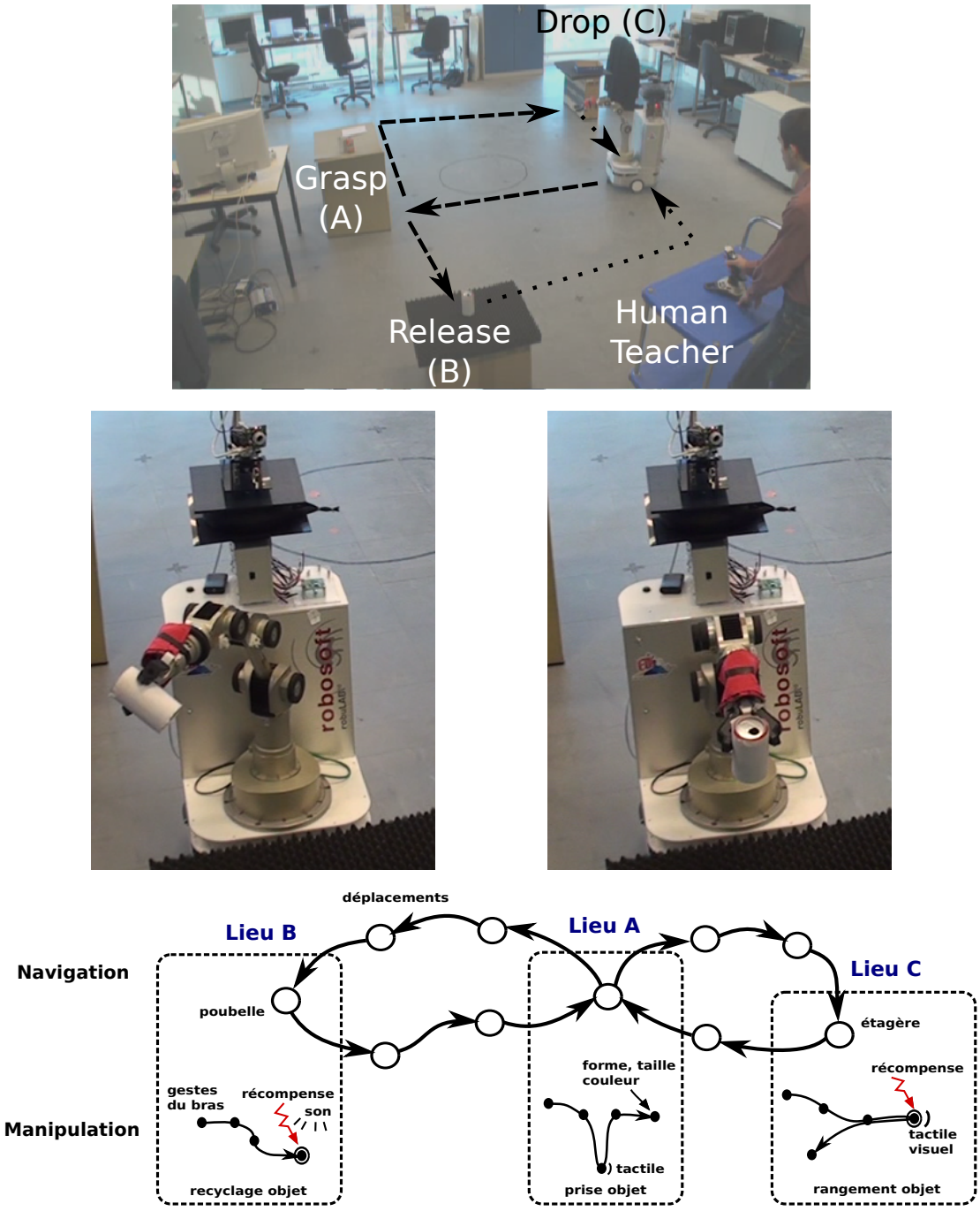


FIGURE 1 – *Haut* : Expérience de tri de canettes avec un robot mobile équipé d'un bras robotique. Le robot navigue entre les différents lieux impliqués dans la tâche et contrôle son bras pour prendre et déposer des objets. *Bas* : Description de la tâche. Les déplacements ou les gestes du bras sont sélectionnés en fonction d'informations multimodales (localisation dans l'espace, son, tactile, posture du bras, etc.) afin de réaliser la tâche. Par exemple, le robot doit s'appuyer sur les informations liées à l'objet (taille, forme,...) pour déterminer dans quel lieu se rendre ensuite. Le lieu où prendre un objet pourrait être fonction de la reconnaissance à distance d'un objet à ranger.

afin de générer des commandes motrices permettant l'action du robot. Les informations visuelles permettent de repérer de loin un objet à ranger mais aussi de se localiser dans la pièce. Les informations tactiles et auditives seront utiles pour déterminer la situation dans laquelle se trouve le robot et donc pour prendre les bonnes décisions sur l'endroit où se rendre ensuite. L'information proprioceptive (liée aux retours sensoriels des actions du robot) sera aussi très importante, notamment pour que le robot puisse s'assurer qu'il suit bien la bonne direction lorsqu'il se déplace ou vérifier qu'il exécute correctement le geste pour atteindre l'objet à déposer. Toutes ces informations peuvent être utilisées, combinées et sélectionnées pour participer à la fois aux contrôles des mouvements de la caméra, des mouvements du bras articulé et de l'orientation du robot durant la navigation. Cela pose la question de savoir comment apprendre ces comportements au cours d'interactions "naturelles" avec un partenaire humain.

Problématique

Permettre à un robot d'apprendre et de reproduire en interaction la tâche proposée (figure 1) soulève de multiples questions. La description écrite de la tâche s'appuie sur une segmentation des différentes actions que le robot sera amené à réaliser. Cependant, un tel découpage, s'il est fixé a priori, fige fortement le type de tâches que pourra apprendre le robot. C'est pourquoi il est souhaitable que le robot puisse découvrir et apprendre à construire les différents niveaux du comportement global qui lui permettra de réaliser une tâche. L'approche développementale en robotique [Lungarella et al., 2003] vise à mettre en place des robots autonomes capables de se développer. Elle s'appuie pour cela sur des études de psychologie du développement. Dans ses travaux, [Piaget, 1936] décrit différentes étapes du développement de l'enfant impliquant l'acquisition de schèmes d'abord sensorimoteurs puis de plus en plus abstraits aboutissant à des capacités de représentation de plus en plus complexes. De la même manière, le comportement global du robot pourrait résulter de comportements frustres liés ensemble selon les situations et les buts, voire les sous-buts, déterminés par le robot. Cette hypothèse soulève plusieurs questions auxquelles j'essaierai d'apporter des réponses dans cette thèse [1].

- Comment peut-on définir un comportement et quel support utiliser pour l'encoder ?
- Comment le système peut-il coordonner les différents comportements pour maintenir la cohérence du comportement global vis à vis de la tâche réalisée ?
- Comment la notion de but peut-elle être associée à la représentation des comportements pour permettre la construction de plans (ou schème [Piaget, 1936]) représentant des tâches complexes ?

Ces questions sont les mêmes pour la construction des comportements de navigation et pour la construction des comportements impliquant le contrôle d'un bras robotique. Existerait-il des mécanismes pour l'apprentissage et l'encodage des comportements qui seraient communs à la navigation et au contrôle d'un bras ?

L'un des enjeux important de cette thèse est l'autonomie du robot. Ainsi, notre robot devrait être capable d'apprendre naturellement d'un professeur durant une interaction non-verbale. Nous entendons par le terme "naturel" l'idée que l'interaction ne devrait pas être différente de celle entre un professeur adulte et un très jeune enfant. Afin d'avoir une interaction dynamique avec

1. Ces travaux ont été réalisés dans le cadre de la fin du projet Européen Feelix Growing et du projet ANR-09-CORD-014 INTERACT.

le robot, nous essaierons de mettre en place un apprentissage rapide et en ligne reposant au maximum sur une architecture réactive. En comparaison des apprentissages hors ligne ou plus long, un apprentissage rapide, et en particulier en un coup, présentera des capacités de généralisation moins bonnes. Suivant une approche incrémentale, nous nous intéresserons à la possibilité dans les solutions proposées d'affiner l'apprentissage pour améliorer les performances.

Dans cette thèse, nous chercherons donc à proposer une architecture capable à la fois :

- de rendre compte des capacités d'interactions physiques et sociales,
- d'apprendre en ligne,
- d'être suffisamment rapide (et donc réactive) pour soutenir une interaction "naturelle" avec un humain,
- d'éviter les représentations symboliques (nous nous situerons à un niveau sub-symbolique, le robot n'aura accès qu'à ses propres perceptions).

L'un des aspects importants de cette thèse est le choix d'une approche développementale et constructiviste appliquée à l'apprentissage de comportements. L'approche constructiviste implique que notre robot (comme tout sujet) ne pourra appréhender la réalité qui l'entoure qu'à travers ses actions et ses capacités d'actions [Maturana and Varela, 1992]. Bien que l'approche développementale n'exige pas de suivre aussi une approche neuromimétique, nous nous intéresserons à des solutions utilisant des réseaux de neurones artificiels biologiquement plausibles. Les neurones pouvant être vus comme des unités de calcul minimales qui permettent "l'émergence" des multiples comportements, la notion d'émergence [2] sera centrale dans cette thèse. Suivant le consensus actuel, une propriété P d'un système sera définie comme émergente *si cette propriété est irréductible aux propriétés microstructurelles du système i.e. si la propriété P ne peut être expliquée et prédite sur la base d'informations relatives aux propriétés des constituants du système* [Sartenaer, 2010]. Cette propriété P est donc constitutive du système et ne peut s'expliquer qu'en tenant compte de la mise en relation des différents composants du système, par exemple la mise en réseau des neurones. Maintenir une contrainte de plausibilité biologique permettra aussi d'avoir un guide dans la mise en place des modèles amenant aux comportements plus sophistiqués.

Dans le chapitre 1, nous présenterons un rapide état de l'art des solutions pour l'apprentissage de tâches en robotique parmi lesquelles l'algorithme LWPR (Locally Weighted Projection Regression), basé sur un régression de modèles locaux linéaires, et l'algorithme GMM/GMR (Gaussian Mixture Model/Gaussian Mixture Regression) basé sur une approche probabiliste du contrôle [Vijayakumar et al., 2005; Calinon et al., 2007]. Puis, nous résumerons très rapidement les données neurobiologiques et psychologiques qui nous seront utiles par la suite.
L'utilisation de modèles cinématiques est une solution classique pour le contrôle visuomoteur cependant ces modèles ne sont pas toujours appris. Les champs de neurones dynamiques (DNF, Dynamic Neural Fields) [Amari, 1977; Schoner, 1995] sont une solution intéressante pour mettre en place un contrôle dynamique robuste de part leur propriété dynamique de bifurcation, de robustesse au bruit et de mémoire. Ils ont été utilisés en navigation [Schoner, 1995] et pour le contrôle d'un bras robotique [Andry et al., 2004; Iossifidis and Schoner, 2004]. Nous présenterons aussi la méthode "Yuragi" de [Fukuyori et al., 2008] qui permettra d'utiliser les attracteurs visuomoteurs existants afin que notre bras de robot atteigne des positions de l'espace d'état qui n'ont pas été explicitement apprises (attracteurs virtuels). Nous discuterons aussi dans quelle mesure les couples sensorimoteurs appris peuvent être réutilisés dans des combinaisons perme-

2. née de la maxime : "le tout est plus que la somme des parties".

ttant d'améliorer la précision des mouvements réalisés.
Nous présenterons ensuite l'architecture PerAc (Perception-Action) [Gaussier and Zrehen, 1995] qui permet l'apprentissage en ligne de comportements sensorimoteurs. Nous comparerons sur un test simple les performances obtenues par LWPR et PerAc. La comparaison montrera que malgré des calculs bien plus frustres pour le modèle PerAc, les performances sur le test effectué sont assez similaires. On constatera que l'algorithme LWPR profite mieux d'un apprentissage à long terme. Cependant, l'algorithme LWPR est aussi légèrement moins performant que PerAc dans les premières itérations de l'apprentissage. Ceci nous confortera dans l'idée que PerAc est particulièrement adapté pour les apprentissages en ligne rapides. Nous illustrerons ensuite l'apprentissage de comportements de navigation en situation d'interaction avec l'approche PerAc [Giovannangeli et al., 2006]. Nous réaliserons alors la comparaison entre PerAc et GMM/GMR dans le cas du suivi d'une trajectoire montrée par un professeur humain.

Dans le chapitre 2, nous montrerons que ce principe d'associations sensorimotrices du modèle PerAc peut être appliqué au contrôle d'un bras via l'apprentissage auto-supervisé d'associations visuomotrices. Nous proposerons dans le modèle Dynamic Muscle PerAc (DM-PerAc) d'associer à une architecture PerAc un modèle de muscles approximés par des ressorts dont la raideur est commandée afin de fournir des capacités de contrôle beaucoup plus fines. Le contrôle dynamique réalisé est équivalent à un contrôle en impédance [Hogan, 1984b] permettant une adaptation simple des mouvements lors des interactions physiques. Dans notre modèle, le recrutement de catégories sensorielles (visuelles et proprioceptives), leur combinaison au sein d'associations multimodales, puis leur association avec des commandes motrices suffira à mettre en place un contrôleur visuomoteur. L'information visuelle et l'information proprioceptive étant fusionnées, le modèle appris sera utilisé comme modèle cinématique direct ou comme modèle cinématique inverse suivant l'influence pondérée de chacune des modalités. Nous montrerons que notre contrôleur DM-PerAc est capable de créer une dynamique dont les propriétés émergentes sont sensiblement les mêmes qu'un champ de neurones dynamiques (DNF) explicite en n'apprenant que quelques couples sensorimoteurs et sans jamais définir un noyau d'interactions latérales comme c'est le cas dans les DNF.

Le chapitre 3 se focalisera sur la coopération Homme-Robot. En situation d'interaction, l'imitation permet à un individu de reproduire les comportements d'autres individus. Dans ses travaux [Piaget, 1945a], Piaget insiste sur l'acquisition de capacités d'imitation différée comme étape-clé dans le développement de l'enfant. Mais le rôle de l'imitation ne peut être restreint à l'apprentissage. Son rôle dans la communication a aussi été mis en évidence [Nadel, 1986]. Gardant à l'esprit l'ensemble de ces idées, je montrerai la capacité de l'architecture DM-PerAc à être employée dans des tâches d'imitation immédiate ou différée, pour apprendre ou communiquer. Nous nous limiterons à des tâches d'imitation permettant d'apprendre de nouveaux comportements à partir de comportements élémentaires donnés ou déjà appris. Nous proposerons une architecture basée sur des apprentissages associatifs simples qui permettra à notre robot d'exhiber à la fois des capacités d'imitation et de planification. Cette architecture complétera notre modèle DM-PerAc par les mécanismes d'apprentissage liés aux cartes cognitives. Dans un premier temps, nous montrerons que DM-PerAc (utilisé comme contrôleur visuomoteur) associé à l'ambiguïté de la perception permet d'obtenir une capacité d'imitation de gestes : si le robot observe la main de son professeur et la prend pour la sienne, il tentera de réduire l'erreur entre ses informations visuelles et proprioceptives induisant un déplacement de son propre bras et mimant ainsi les mouvements du bras humain.

Les positions visuelles ou proprioceptives apprises peuvent ensuite être utilisées pour la création de transitions permettant de mémoriser des séquences visuelles ou proprioceptives. La transition à réaliser viendra biaiser l'activation des catégories visuomotrices et ainsi sélectionner les mouvements appropriés pour reproduire la séquence encodée. Durant une démonstration visuelle, le robot devra inhiber ses propres mouvements afin de ne regarder que les gestes du professeur. Nous montrerons alors que notre contrôleur visuomoteur DM-PerAc exploitant des transitions est suffisant pour apprendre par observation une séquence visuelle et la reproduire. Notre robot pourra ainsi exhiber des comportements d'imitation différée.
Afin de gérer les différentes séquences apprises, notre modèle DM-PerAc sera complété par un mécanisme d'apprentissage lié aux cartes cognitives utilisées en navigation [Tolman, 1948; Banquet et al., 1997; Gaussier et al., 2002; Cuperlier et al., 2006]. Les transitions réalisées seront associées ensemble pour former un graphe des transitions successives. A la réception d'une récompense, la dernière transition réalisée sera associée dans le graphe avec le contexte décrivant la motivation actuelle du robot. Ce contexte motivationnel est lié à l'état du robot dans lequel il pourra chercher à obtenir une récompense le satisfaisant [3] (e.g. si le robot a faim (drive), il cherchera de la nourriture pour manger (récompense)). Chez un humain, les choix réalisés sont bien sûr dépendant de la situation telle qu'elle est perçue mais aussi de certains états internes comme l'état émotionnel. L'environnement social peut jouer un rôle important dans la prise de décision en influençant cet état émotionnel.
Nous nous appuierons sur cette dimension sociale pour extraire un signal de renforcement permettant de moduler les comportements du robot dans le cas d'une interaction. Nous montrerons que la carte cognitive est une solution adéquate pour apprendre à résoudre un problème de tri de canettes suite à une manipulation passive du robot. Après apprentissage, le robot planifiera la séquence de gestes à réaliser en fonction de la couleur de la canette attrapée. Notre modèle permettra aussi d'adapter le comportement du robot grâce aux signaux de renforcement positifs ou négatifs issus de l'interaction sociale, enlevant la contrainte de devoir toujours montrer la séquence de gestes désirée. Enfin, grâce à la carte cognitive, le robot pourra aussi rapidement replanifier ses mouvements.

Pour finir, dans le chapitre 4, nous nous intéresserons à la sélection de comportements dans le cadre d'une tâche impliquant des comportements de navigation et de contrôle d'un bras. L'existence de plusieurs comportements réalisables à un instant donné nécessite que notre robot soit capable de sélectionner le comportement le plus approprié en fonction de la situation et de la tâche qu'il doit réaliser. Une sélection est ainsi nécessaire lorsqu'il y a différentes manières d'apprendre et de reproduire une tâche (apprentissage par observation versus apprentissage par manipulation passive par exemple). Cette sélection est d'autant plus nécessaire quand la tâche considérée implique différents comportements de navigation et de contrôle du bras.
Nous commencerons par observer, à travers une expérience sur robot, que la synergie de comportements s'exécutant en parallèle peut suffire pour faire émerger des comportements relativement complexes. Notre robot mobile, s'orientant vers un objet et alignant visuellement sa pince sur ce même objet sera capable de se déplacer pour attraper un objet sans qu'aucun processus de synchronisation ni planification ne soit nécessaire. Cependant, une telle synergie ne peut suffire

3. Dans une approche animat [Meyer, 1996] de la navigation, le système est guidé par ses drives (contextes motivationnels) correspondant à des besoins physiologiques devant être satisfaits. La satisfaction d'un besoin est directement perçue comme une récompense par le cerveau e.g. manger quand on a faim fait monter le niveau de glucose dans le sang.

pour résoudre toutes les tâches. Les comportements doivent parfois être exécutés indépendamment, ce qui implique leur sélection par le système.
Nous nous intéresserons à l'inhibition de comportements par l'inhibition des couples sensorimoteurs correspondant. Nous proposerons pour cela un mécanisme d'évaluation des couples sensorimoteurs qui s'appuie sur les actions du professeur humain lorsqu'il corrige le comportement du robot. L'évaluation d'un couple sensorimoteur sera réalisée à partir de la comparaison entre les changements perceptifs attendus pour ce couple et ceux produits par l'action du professeur. Cette évaluation déterminera ainsi les couples sensorimoteurs indésirables qui seront alors associés avec des contextes recrutés au moment de la correction. Par la suite, ces couples seront inhibés si la même situation se représente (contexte actif). Nous validerons ce modèle sur l'apprentissage de différentes trajectoires dépendant d'un contexte (e.g. type d'objet tenu).
Enfin, nous compléterons notre étude par l'utilisation de contextes renforçant l'activation de certains comportements et biaisant ainsi leur sélection. Nous présenterons une expérience réalisée en collaboration avec le laboratoire LASA de l'EPFL dans laquelle nous avons intégré ensemble leur solution basée sur des mixtures de Gaussiennes [Calinon et al., 2007] pour le contrôle d'un bras avec la solution PerAc pour la navigation du robot. Cette expérience, qui consiste en une tâche de tri avec déplacement entre différents lieux (prise d'un objet, déplacement selon l'objet, dépôt, poursuite de la navigation pour prendre un autre objet, Figure 1), sera apprise par notre robot grâce à notre modèle de sélection de comportements.

Enfin, dans le dernier chapitre, une discussion finale rappellera les contributions et les limitations de ce travail, ainsi que les perspectives envisagées.

Il n'y a d'autre savoir que de savoir qu'on ne sait rien, mais on ne le sait qu'après avoir tout appris.

– Maurice Chapelan

CHAPITRE 1

État de l'art

Il est difficile de produire un état de l'art donnant un aperçu exhaustif de l'ensemble des aspects qui touchent au contrôle moteur et à son implication dans l'apprentissage de comportements. Dans cette partie, je vais donc présenter les travaux qui m'ont parus fondamentaux pour la mise au point de mon architecture de contrôle sensorimoteur.

Dans un premier temps (Sec. 1.1), j'introduirai la problématique du contrôle moteur sous l'angle du contrôle optimal. Suivant le double objectif de comprendre le contrôle moteur humain et de réaliser le contrôle de robots, il s'agit de l'approche qui a reçu et reçoit encore le plus d'attention. Cette approche nous permettra d'aborder les différents enjeux du contrôle moteur : la capacité à converger, les critères d'adaptation ou d'apprentissage, et les critères d'optimisation et de généralisation du modèle.

Dans un second temps (Sec. 1.1.4) je montrerai dans cet état de l'art qu'une vision purement "optimisation des paramètres du contrôleur" peut être en décalage avec les résultats et les spéculations récentes sur la manière dont l'Homme effectue son contrôle moteur, avec notamment un effet du contexte social sur les paramètres bas niveau de contrôle de la tâche : ainsi la notion d'optimisation peine à rendre compte de l'influence du contexte social sur l'adaptation en ligne des mouvements humains. Nous discuterons donc les limites du contrôle optimal pour expliquer la construction de comportements en particulier dans le cadre d'interactions sociales.

Ce constat m'amènera naturellement à étudier et à présenter les travaux récents sur le contrôle sensori-moteur et l'apprentissage chez le vivant, et notamment les structures cérébrales impliquées (Sec. 1.2). Si une partie des structures cérébrales semble très liée aux aspects d'optimisation, d'autres principes implémentés dans le cerveau nous dirigent vers les approches se focalisant sur la dynamique des comportements et leur modélisation (Sec. 1.3).

Je présenterai ensuite (Sec. 1.4) les travaux précédents du laboratoire en navigation qui ont inspiré ma démarche. Indépendamment du contexte expérimental, ces travaux introduisent les principes fondamentaux de mon modèle d'apprentissage d'associations sensorimotrices pour la construction d'attracteurs comportementaux (cf. chapitre 2).

Enfin, je ferai le lien avec les comportements d'imitation, qui sont pour moi à la fois une conséquence et une cause du développement d'un agent en interactions physiques et sociales permanentes avec son environnement (Sec. 1.5).

1.1 Théorie du contrôle optimal et apprentissage

La théorie du contrôle optimal est un champ de l'automatique qui a donné lieu à de nombreuses applications. En particulier, cette théorie a beaucoup été utilisée pour expliquer le contrôle de mouvements biologiques et notamment la locomotion, le contrôle oculaire, les mouvements de la main et même du corps dans son ensemble. Cette théorie présente deux avantages principaux : elle permet de réaliser le contrôle de robots et elle peut aussi être appliquée au contrôle moteur chez l'humain pour interpréter les propriétés observées [Ivaldi et al., 2012]. Je ne présenterai pas ici un état de l'art exhaustif du contrôle optimal, je viserai principalement à introduire ses avantages et ses défauts.

Dans cette section, je commencerai par donner les principes fondamentaux de la théorie du contrôle optimal : le principe d'optimalité de Bellman [Bellman, 1957] et le principe du maximum de Pontryagin [Pontryagin et al., 1962]. Je donnerai ensuite une idée des différentes problématiques liées au contrôle optimal : résolution discrète ou continue, choix de la fonction de coût, prise en compte du bruit, contrôle en boucle ouverte ou en boucle fermée, et la prise en compte de l'apprentissage. Enfin, je présenterai les limites de cette théorie dans le cas de comportements d'interactions sociales.

1.1.1 Principes du contrôle optimal

Soit $x \in \mathcal{X}$ l'état d'un agent dans un environnement donné et $u \in \mathcal{U}$ l'action (ou le contrôle) que l'agent peut choisir d'employer dans l'état x. On introduit alors l'idée que la réalisation de u dans l'état x a un coût pour le système $cout(u, x)$. Suivant des contraintes sur l'état initial et l'état d'arrivée, la séquence $(x_{1..n}, u_{1..n})$ optimale sera celle avec le coût total $J(x_., u_.) = \sum_{k=0}^{n-1} cout(x_h, u_k)$ le plus faible. En discret, la programmation dynamique (DP, Dynamic Programming) résout ce problème d'optimisation grâce au principe d'optimalité de Bellman. On introduit alors une fonction v de valeur optimale définissant le coût sur la séquence optimale d'action à partir d'un état donné x.

$$v(x) = \text{“coût le plus faible pour réaliser la tâche à partir de } x\text{”} \tag{1.1}$$

On définit alors la loi de contrôle optimal π par :

$$\pi(x) = \underset{x \in \mathcal{U}(x)}{argmin}\{cout(x, u) + v(suivant(x, u))\} \tag{1.2}$$

avec $suivant(x, u)$ l'état dans lequel se retrouve le système en réalisant u à partir de x. La fonction de valeur v est définie par :

$$v(x) = \underset{u \in \mathcal{U}(x)}{min}\{cout(x, u) + v(suivant(x, u))\} \tag{1.3}$$

Ces deux équations sont les équations de Bellman. Le principe d'optimalité peut être appliqué au cas stochastique en tenant compte d'une loi de probabilité sur l'état d'arrivée quand on réalise l'action u dans l'état de départ x. Dans ce cas, on adaptera les équations de Bellman en prenant l'espérance de la fonction de valeur. D'autre part, ce principe a aussi été étendu au cas continu. Les équations utilisées sont alors les équations de Hamilton-Jones-Bellman.

La seconde façon d'aborder la théorie du contrôle optimal est le principe du maximum de Pontryagin. En résumé, il s'agit de simplifier les équations en utilisant la dérivation et en remarquant que la dérivée partielle suivant u de la quantité que l'on minimise (ou maximise) selon u sera nulle. On simplifie les équations et on aboutit à une équation différentielle ordinale que l'on peut résoudre. Le principe du maximum ne s'applique qu'au cas déterministe et il est alors équivalent à la programmation dynamique. Selon les situations, on peut donc utiliser l'une ou l'autre approche.

Dans [Todorov, 2006],l'auteur présente en détail les démonstrations permettant de construire les solutions optimales à partir de ces deux principes. Les problèmes de contrôle optimal dépendent de nombreux facteurs : le contrôle dynamique utilisé, la fonction de coût, le type de bruit présent ou non. Dans le cas général, leur résolution implique souvent des calculs itératifs, parfois très coûteux (malédiction de la dimension [1]), pour obtenir une solution approximée. Il existe cependant des cas comme le contrôle linéaire quadratique gaussien (LQG, *Linear Quadratic Gaussian*) où une solution analytique peut être calculée. Il s'agit du cas où la dynamique du contrôle est linéaire, où les coûts sont quadratiques (action et état) et où le bruit est additif gaussien (si présent). Dans le cas déterministe (pas de bruit), on parle aussi de régulateur quadratique linéaire (LQR, *Linear Quadratique Regulator*). La solution de ce problème est connue en fonction des paramètres de contrôle et de coût et donc ce modèle est souvent utilisé.

1.1.2 Diversité des contrôleurs optimaux : choix de la fonction de coût et choix du type de contrôleur

Nous avons vu dans la section précédente que le contrôle optimal s'appuie sur la minimisation d'une quantité (fonction de valeur v). De ce fait, le contrôle que l'on obtient sera dépendant de la quantité que l'on va minimiser. Les modèles classiques de contrôle optimal en robotique s'appuient sur la minimisation de caractéristiques du mouvement pour générer une trajectoire désirée en boucle ouverte. Les caractéristiques minimisées peuvent être liées à la consommation d'énergie, au lissage de la trajectoire ou encore à la précision obtenue sur le mouvement dans le cas stochastique. Les modèles minimum d'énergie s'appuient sur l'idée que les mouvements générés par les humains visent à réduire le coût énergétique pour les muscles [Hatze and Buys, 1977]. Différents modes de calcul sont possibles pour estimer l'énergie consommée : le calcul peut être basé sur le couple [Nelson, 1983] ou bien sur un modèle du muscle plus précis [Taniai and Nishii, 2008; Nishii and Taniai, 2009]. Le principe d'inactivation proposé dans [Berret et al., 2008; Gauthier et al., 2010] vise à minimiser les périodes de co-activation musculaire.
Afin de générer une trajectoire lisse, différents critères peuvent être minimisés comme les secousses (i.e. la dérivée de l'accélération de la main) (*minimum-jerk*, [Hogan, 1984a; Flash and Hogan, 1985]) et la dérivée du couple (*minimum torque change*, [Uno et al., 1989; Nakano et al., 1999]). Ce type de modèle permet d'obtenir des mouvements qui reproduisent assez fidèlement la dynamique des mouvements d'atteinte humains (trajectoire droites, profil de vitesse en cloche).
Une autre grande classe de modèle de contrôle optimal vise à minimiser l'erreur en réduisant la variance. Le modèle *minimum variance* de Harris et Wolpert [Harris and Wolpert, 1998] cherche à réduire la variance sur la position de l'extrémité du bras à la fin du geste. Le contrôle réalisé est

1. *curse of dimensionality*, identifiée par R. Bellman lorsqu'il travaillait sur des problèmes d'optimisation dynamique [Bellman, 1957]

fait en boucle ouverte. Une version de ce contrôle en boucle fermé est proposée par Todorov et Jordan [Todorov and Jordan, 2003]. On aboutit ainsi au principe de l'intervention minimale. Un écart par rapport à la trajectoire initiale ne sera corrigé que s'il a une influence sur la précision à la fin du mouvement. Le contrôle peut ainsi libérer les contraintes sur les degrés de liberté qui n'ont pas d'influence sur la précision finale.
Ces différents modèles permettent d'expliquer certaines propriétés du mouvement (relation vitesse-curvature, trajectoire droite, profils en cloche de la vitesse, etc.) dans certaines situations. Cependant, il n'y a pas de modèle qui soit le meilleur dans toutes les situations. Ainsi, il apparaît qu'une combinaison de ces différentes approches soit plus correcte. Cette combinaison pourrait aussi évoluer en fonction des situations.

Le second point important dans le contrôle optimal est le type de contrôleur utilisé. Dans [Todorov, 2004], l'auteur insiste en particulier sur le choix entre contrôleur en boucle ouverte et contrôleur en boucle fermée. Dans le cas d'un contrôle en boucle ouverte, le contrôle optimal revient à calculer une trajectoire désirée. L'inconvénient du contrôle en boucle ouverte est le manque de flexibilité et d'adaptation du contrôleur. Une solution est d'utiliser cette trajectoire désirée comme une commande pour un contrôleur moteur bas niveau tel qu'un contrôleur en impédance permettant d'introduire de la souplesse et de l'adaptation dans les commandes générées.

Contrôle en impédance

Des études sur les propriétés du mouvement ont conduit au modèle du contrôle en impédance [Hogan, 1984a] en tant qu'approximation des propriétés neuro-musculaires. Le contrôle en impédance est efficace pour contrôler des manipulateurs agissant en contact physique avec l'environnement [Chiaverini et al., 1999]. C'est une approche usuelle pour le design de prothèses et d'exosquelettes qui impliquent une interaction physique directe avec un humain [Jiménez-Fabián and Verlinden, 2011]. Le contrôle en impédance est basé sur un système dynamique du second ordre de type "masse ressort amorti" (1.4) qui permet de gérer les mouvements contraints, les interactions dynamiques avec l'environnement et l'évitement d'obstacles.

$$M\frac{dV}{dt} = K(X_0 - X) + B(V_0 - V) \tag{1.4}$$

avec M la masse, V la vélocité et X la position cartésienne de l'extrémité du bras [2]. Les coefficients K (équivalent à une raideur de ressort) et B représentent les contributions respectives des contraintes liées à la commande en position X_0 et à la commande en vitesse V_0. Ce contrôle en impédance peut être utilisé avec une méthode d'optimisation en tenant compte de contraintes de points de passage (*via-points*) pour générer des trajectoires plus complexes qu'un simple mouvement d'atteinte. Selon l'hypothèse de la trajectoire à l'équilibre (*equilibrium trajectory*, [Flash, 1987]), la trajectoire désirée utilisée dans le contrôle en impédance est obtenue à partir de la minimisation des secousses. Les points de passage nécessaires pour calculer la trajectoire optimale correspondante peuvent être découverts par observation [Miyamoto and Kawato, 1998]. Dans ce cas, les points sont dans l'espace cartésien et obtenus à partir de caméras calibrées. Il est possible d'adapter automatiquement le timing pour le passage aux différents points de passage en fonction de la durée allouée pour le mouvement. Le modèle proposé dans [Wada and Kawato, 2004] permet ainsi d'adapter les mouvements de l'écriture manuelle réalisée à différentes vitesses.

2. On notera que certaines versions du contrôle en impédance utilisent l'information proprioceptive articulaire plutôt que la position Cartésienne (e.g. [Albu-Schäffer et al., 2007]).

Néanmoins, les capacités d'adaptation aux changements sensoriels et aux perturbations de ces contrôleurs en impédance restent limitées. Le modèle Optimal Feedback Control (contrôle optimal en boucle fermée) présenté dans [Todorov and Jordan, 2002] permet de prendre en compte le retour sensoriel. La fonction de coût utilisée s'appuie sur un critère énergétique et un critère de précision aboutissant au principe d'intervention minimale. Ainsi, même si le système s'écarte de la trajectoire initialement prévue, le système va adapter son mouvement pour finir le contrôle de manière optimale et non essayer de revenir sur la trajectoire prévue. La mise en place d'un tel contrôle optimal en boucle fermée s'appuie sur la définition d'une correspondance sensorimotrice qui est complexe et coûteuse à calculer a priori, ou qui nécessite d'avoir un modèle précis du système utilisé. La théorie de contrôle optimal est aussi lié à différentes solutions pour apprendre à réaliser des tâches.

1.1.3 Apprentissage par démonstration et apprentissage par renforcement

En discret, le principe d'optimalité a conduit à l'apprentissage par renforcement [Sutton and Barto, 1998]. Suivant le principe d'une valence propagée pour évaluer les performances et sélectionner la meilleure action, les algorithmes de Q-learning [Watkins and Dayan, 1992] et de TD lambda (Temporal Difference) peuvent être utilisés pour réaliser un contrôle moteur. Les limites sont que le contrôle est généralement fait dans un espace d'état et d'action discrétisé. Afin de construire le modèle de ses capacités et de sélectionner quelle action réaliser dans quelle circonstance, le robot peut réaliser un babillage moteur (exploration aléatoire de capacités motrices). Dans un paradigme d'apprentissage par renforcement, la phase d'exploration pour permettre au système d'approcher une bonne solution dans un espace d'état de grande dimension peut être très longue. *Les démonstrations faites par un humain peuvent permettre d'accélérer la phase d'exploration d'un apprentissage par renforcement* en réduisant le nombre d'états à explorer. Dans le but d'apprendre comment accomplir une tâche, un robot peut utiliser des données ou un retour fourni par un professeur humain et intégrés dans un modèle de la tâche [Schaal, 1997]. Il peut s'agir d'un modèle de la dynamique permettant d'avoir un meilleur contrôle des mouvements dès le départ. Il peut aussi s'agir d'un modèle du monde utilisé comme valeurs initiales pour le Q-learning. Grâce à ces modèles, une simulation mentale peut être réalisée pour poursuivre l'apprentissage sans nécessiter la réalisation les mouvements.

Le contrôle optimal peut aussi être défini par rapport aux données récupérées lors d'une démonstration faite par un professeur humain. Ainsi, le coût correspond à la qualité de la reproduction des gestes montrés par l'humain. Je présenterai en particulier l'approche Dynamic Motion Primitives ainsi que l'approche Régression de Mixtures de Gaussiennes.

Apprentissage local incrémental et régression de fonctions

L'approximation de fonctions basée sur des techniques de régression locales [Atkeson et al., 1997] est efficace pour l'apprentissage de modèles directs ou inverses du contrôle d'un robot. Dans [Schaal and Atkeson, 1998], les auteurs ont proposé l'algorithme RFWR (Receptive Fields Weighted Regression, Régression Pondérée sur Champs Récepteurs) pour l'apprentissage incrémental local d'approximations de fonctions. Il s'appuie sur une régression pondérée de modèles linéaires paramétriques associés à des champs récepteurs (RF). La pondération est fonction de la taille et de la forme du champ récepteur qui répond à une entrée suivant une fonction noyau (gaussienne) paramétrée (centres, matrices de covariance). Les paramètres de la fonction linéaire

sont obtenus par une méthode récursive des moindres carrées sur les données fournies[Ljung and Söderström, 1983]. Les paramètres des fonctions noyaux sont mis à jour par une descente de gradient récursive prenant en compte une validation croisée sur les prédictions et une pénalité sur le nombre de RF et leur recouvrement. Un nouveau champ récepteur est recruté quand aucun RF ne répond suffisamment à une entrée donnée. Il est aussi possible d'élaguer les champs récepteurs lorsqu'ils se recouvrent trop. Développé récemment, l'algorithme de Locally Weighted Projection Regression (LWPR, Régression et Projection Localement Pondérée) [Vijayakumar et al., 2005] fusionne les propriétés d'apprentissage incrémental de l'algorithme RFWR et une projection des données d'entrée afin de réduire la dimensionnalité du problème. Les auteurs ont réalisé des expériences avec un robot humanoïde SARCOS avec 30-DDL apprenant le modèle dynamique inverse et reproduisant des trajectoires en forme de huit avec l'extrémité de son bras. Le modèle LWPR a aussi été utilisé pour apprendre un modèle cinématique direct pour le contrôle visuomoteur [Natale et al., 2007; Droniou et al., 2012] d'un bras robotique. Le contrôle nécessite un traitement supplémentaire pour obtenir le modèle inverse comme par exemple le calcul de la pseudo inverse de la Jacobienne [Droniou et al., 2012].
Les techniques de régression pour apprendre des modèles de contrôle moteur ont aussi été utilisés dans le paradigme de l'apprentissage par démonstration [Argall et al., 2009]. L'algorithme de Dynamic Movement Primitives (DMP, Primitives Dynamiques de Mouvement) [Ijspeert et al., 2003; Schaal, 2003; Hoffmann et al., 2009] est basé sur l'algorithme RFWR. Les primitives sont des stratégies de contrôle activées par une fonction de base locale et contribuent au système dynamique du second ordre générant le contrôle moteur. La combinaison des primitives donne forme au paysage attracteur pour produire la trajectoire désirée. La fonction approximée est la trajectoire alignée sur une variable de phase variant de 0 à 1 plutôt que sur le temps réel. La régression locale pondérée des données d'entraînement détermine les paramètres des fonctions base (nombre, centre, largeur) et la contribution des primitives correspondantes. L'algorithme DMP montre des propriétés intéressantes d'invariance spatiale et temporelle et a été appliqué à l'apprentissage des mouvements rythmiques et discrets. Cependant, la dépendance de la trajectoire à la variable de phase limite aussi la robustesse aux perturbations. Dans [Ijspeert et al., 2001, 2003], les données d'entraînement sont obtenues par l'enregistrement des angles articulaires chez le professeur humain impliquant un robot humanoïde contrôlé avec une morphologie proche de celle d'un humain.

Modèles de mélange probabilistes

De manière similaire, le modèle GMM (Gaussian Mixture Model) peut aussi apprendre un modèle de tâches démontrées en apprenant les probabilités conjointes entre les informations proprioceptives et cartésiennes sous la forme de noyaux gaussiens adaptés [Calinon et al., 2007]. L'apprentissage est basé sur l'algorithme d'Expectation-Maximization (Table 1.1) adaptant les noyaux gaussiens pour qu'ils décrivent de manière probabiliste les données d'entrée obtenues lors de la session d'entraînement. Ensuite, étant donné des informations partielles comme par exemple uniquement la position cartésienne, une régression sur les mixture de gaussiennes (GMR, Gaussian Mixture Regression) permet d'extraire la proprioception qui doit être utilisée pour commander le bras robotique. Selon la tâche, un système de vision ou de capture du mouvement peut suivre des éléments particuliers (par exemple la cuillère et la tête de l'humain pour apprendre à manger avec une cuillère) afin de fournir les données d'entrée à l'algorithme [Calinon et al., 2010a,b]. Le calcul des coordonnées cartésiennes des marqueurs visuels nécessite la calibration

TABLE 1.1 – Algorithme GMM/GMR (Gaussian Mixture Model/Gaussian Mixture Regression)

Un modèle à mixture de gaussiennes (GMM : Gaussian Mixture Model) définit une fonction de densité de probabilité $p(\xi)$ sur l'espace d'état du robot avec ξ la position du robot dans cet espace.

$$p(\xi) = \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} e^{-\frac{1}{2}\left((\xi-\mu_k)^\top \Sigma_k^{-1}(\xi-\mu_k)\right)} \tag{1.5}$$

D est la dimension de l'espace d'état et K est le nombre d'états. Les valeurs π_k sont les probabilités a priori que la position ξ soit due à l'état k, les μ_k sont les centres des noyaux gaussiens et les Σ_k représentent les matrices de covariance. Étant donné un enregistrement d'entraînement contenant n points (ξ_i), un algorithme de Expectation-Maximization (EM) est utilisé pour trouver les paramètres π_k, μ_k, Σ_k qui maximisent la vraisemblance (1.7) pour cet enregistrement.
Expectation : Dans un premier temps, on estime la probabilité pour chaque point (ξ_i) de l'enregistrement pour qu'il soit généré par un noyau gaussien (état k) $p(k|\xi_i)$.

$$p(k|\xi_i) = \frac{p(\xi_i|k)p(k)}{\sum_{j=1}^{K} p(\xi_i|j)p(j)} \tag{1.6}$$

avec $p(\xi_i|k) = \mathcal{N}(\xi_i, \mu_k, \Sigma_k)$, $\mathcal{N}$ étant la loi normale, et $p(k) = \pi_k$. La vraisemblance $\mathcal{L}(x)$ correspondant à la possibilité d'observer cet ensemble de données d'entraînement compte tenu des paramètres du modèle à l'itération x de l'algorithme d'EM est donnée par :

$$\mathcal{L}(x) = \prod_{i=1}^{n} \left(\sum_{k=1}^{K} p(\xi_i|k)\pi_k \right) \tag{1.7}$$

C'est cette équation qui sert de mesure de convergence et de performance pour l'algorithme afin de savoir si l'apprentissage du modèle est satisfaisant et achevé.
Maximization : La moyenne μ_k et la matrice de covariance Σ_k pour chaque noyau gaussien k sont recalculées en pondérant chaque donnée par sa probabilité $p(k|\xi_i)$ d'être dans ce noyau k. Les deux étapes d'***Expectation*** et de ***Maximization*** sont réitérées jusqu'à convergence.
La régression de mixture de gaussiennes (GMR : Gaussian Mixture Regression) permet d'obtenir des informations partielles manquantes $\xi^{\mathcal{O}}$ à partir d'un information partielle connue $\xi^{\mathcal{I}}$ en s'appuyant sur les statistiques extraites par le modèle. Soit $\xi = [\xi^{\mathcal{O}} \xi^{\mathcal{I}}]$ un point de l'espace d'état issu de la concaténation de la sortie $\xi^{\mathcal{O}}$ recherchée et d'une entrée $\xi^{\mathcal{I}}$ connue. La GMR permet alors d'estimer $\xi^{\mathcal{O}}$ en prenant la moyenne donnée par la distribution de l'attente (1.8) (se reporter à [Cohn et al., 1996] pour plus de détails).

$$p(\xi^{\mathcal{O}}|\xi^{\mathcal{I}}) \sim \sum_{k=1}^{K} p(k|\xi^{\mathcal{I}})\ p(\xi^{\mathcal{O}}|\xi^{\mathcal{I}}, k), \tag{1.8}$$

des différents appareillages. Dans [Calinon et al., 2009], l'algorithme initial basé sur le GMM a été modifié pour d'une part utiliser un contrôleur moteur avec une dynamique du second ordre et d'autre part exploiter des Modèles de Markhov Cachés (HMM, Hidden Markhov Models) pour remplacer le GMM. Les HMM peuvent prendre en compte les dépendances séquentielles dans une tâche tandis que le contrôle moteur du second ordre réalise un contrôle en impédance. Un compromis entre la contrainte en position et la contrainte en vitesse est géré selon la variabilité dans les trajectoires démontrées. Cette version du modèle est similaire aux DMP. La différence majeure est que l'apprentissage des contraintes en position et en vitesse peut prendre en compte l'influence mutuelle entre les différents degrés de liberté, contrairement à l'apprentissage fait dans DMP.

Ces différentes solutions s'appuient sur des apprentissages capables d'extraire les propriétés statistiques de la tâche démontrée. Afin d'obtenir les données d'entraînement nécessaires, différentes démonstrations entièrement guidées doivent être réalisées ce qui est assez contraignant pour le professeur humain. D'autre part, l'apprentissage en lui-même est généralement réalisé hors ligne tandis que les données enregistrées durant les démonstrations sont rejouées en interne. Certaines solutions basées sur des mixtures de gaussiennes permettent toutefois un apprentissage plus incrémental, comme par exemple [Calinon and Billard, 2007; Cederborg et al., 2010].

1.1.4 Critique du contrôle optimal : le cas des mouvements humains dans un contexte social

Si l'approche du contrôle optimal a permis d'expliquer beaucoup de propriétés cinématiques des mouvements humains, il existe cependant des situations où elle se trouve en défaut. La problématique principale du contrôle optimal est de s'appuyer sur la fonction de coût que l'on essaie de minimiser lors des mouvements produits. Cette fonction de coût est donnée a priori. Elle suppose donc de pouvoir extraire objectivement les paramètres du comportement moteur qui doivent être optimisés.

Il existe des situations où les mouvements apparaissent comme suboptimaux au vu des paramètres classiquement optimisés. Cela laisse apparaître des paramètres difficiles à inclure dans la fonction d'optimisation et qui ont pourtant une influence sur la cinématique du mouvement produit. Les mouvements d'approche pour saisir un objet montrent des caractéristiques cinématiques différentes (vitesse, préparation de la prise) selon les mouvements à réaliser ensuite avec l'objet (e.g. déplacer pour insérer dans des emplacements plus ou moins larges) [Ansuini et al., 2006]. Ainsi, afin d'augmenter le confort à la fin du mouvement et de faciliter l'action suivante, le mouvement peut être sous-optimal sur la partie initiale de la trajectoire [Rosenbaum and Jorgensen, 1992; Short and Cauraugh, 1997, 1999]. Le contexte social a aussi une influence sur les comportements réalisés. L'expérience de pensée "docteur Jekyll et Mr. Hyde" proposée par [Jacob and Jeannerod, 2005] pose le problème suivant : à la vue de l'enregistrement de la trajectoire d'un scalpel, est-il possible de savoir quelle est l'intention du praticien ? S'agit-il d'un médecin en train de sauver son patient ou d'un psychopathe en train de prendre un plaisir sadique à la souffrance de sa victime ? En d'autre termes, est-il possible d'interpréter l'intention (e.g. soigner ou faire souffrir) à partir de l'observation des mouvements réalisés ? Selon les auteurs, des mouvements issus d'une même intention motrice (un même objectif moteur e.g. faire une incision suivant une certaine procédure) sont identiques et ne peuvent donc pas permettre de différencier les intentions sociales (le docteur versus le sadique). Cependant, l'a priori fort

que les mouvements ne dépendraient que de l'intention motrice et non des intentions sociales est contredite par certains résultats expérimentaux. Dans l'expérience réalisée dans [Georgiou et al., 2007], les sujets doivent attraper un objet toujours posé au même endroit, et le déplacer à un endroit prédéfini. Quatre conditions sont testées : dans les conditions où le sujet est seul ou avec un observateur passif, le sujet réalise son mouvement sans interaction. Dans une condition de coopération, deux sujets doivent poser leur bloc l'un sur l'autre en présence ou non d'un observateur. Dans la dernière condition dite de compétition, les deux sujets essayent de poser leur bloc en premier. Les comparaisons sont effectuées sur le mouvement initial d'atteinte pour la prise de l'objet. Les auteurs ont observé chez des humains que les mouvements réalisés seuls, ou avec un observateur passif, sont différents des mouvements réalisés en situation de coopération, eux même encore différents de ceux réalisés en situation de compétition. Les mouvements d'atteinte montrent des cinématiques différentes selon les conditions de l'expérience. Afin d'écarter une possible influence mutuelle entre les mouvements des deux sujets, dans l'expérience de [Becchio et al., 2008], un seul sujet réalise un mouvement à un instant donné. Les deux conditions sont (1) poser l'objet pris sur une zone prédéfinie ou (2) poser l'objet sur la main d'un humain afin qu'il réalise ensuite un geste avec l'objet. Le simple fait de donner un objet à un humain au lieu de le poser sur une table influence toute la trajectoire, y compris la partie du mouvement pour attraper l'objet. Les cinématiques des mouvements ont été modifiées par la présence ou non de l'observateur au point de dépôt. Le mouvement semble donc impacté par le contexte social. Cette expérience a été répliquée dans [Lewkowicz et al., 2013] avec des résultats similaires. De plus, Lewkowicz et collègues montrent que les caractéristiques cinématiques du mouvement sur la phase d'approche de l'objet sont suffisamment discriminantes pour permettre à un humain de reconnaître la nature du contexte (social ou pas) avec une probabilité supérieure au hasard. Un simple réseau neuronal est même capable d'apprendre le contexte à partir de ces caractéristiques.

Ainsi, une action ne peut se limiter à une optimisation de paramètres moteurs en dehors de tout contexte psychologique et social. L'influence de l'intention sociale ou préalable apparaît à très bas niveau dans la réalisation motrice. Les solutions basées sur une optimisation ou un apprentissage statistique tiennent bien compte de la généralisation à différents buts de manière dynamique (intention motrice). Cependant, ces solutions apparaissent assez limitées pour la prise en compte d'une modulation des gestes par le contexte social et les motivations du système. Contrairement à un modèle hiérarchique où les contextes seraient considérés à un niveau supérieur presque indépendant du contrôle moteur bas niveau, il nous semble important de considérer la possibilité de liens forts entre tous les niveaux d'encodage des comportements. On notera cependant que le modèle de [Todorov and Jordan, 2003] du minimum d'intervention peut apporter un éclairage intéressant pour interpréter l'influence sociale. Dans les expériences mentionnées, le fait que les actions réalisées par le partenaire puissent difficilement être prédites peut expliquer la modification des mouvements en situation d'interaction. Ainsi, la contrainte serait la variabilité des mouvements du partenaire plus que la dimension sociale en tant que telle. Cependant, une telle interprétation sous estime l'aspect attentionnel de la tâche. Une exagération des gestes peut permettre de récupérer l'attention du partenaire et faciliter la collaboration sur la réalisation d'une tâche. Lorsque un humain veut enseigner quelque chose à un robot, les capacités d'attention du robot, et en particulier son regard, influenceront l'amplitude et la forme des des gestes de l'humain [Nagai et al., 2010; Thomaz and Cakmak, 2009]. Agir pour attirer l'attention du partenaire pourrait être considéré comme une stratégie pour augmenter la prédictibilité des gestes du

partenaire. Bien que l'on puisse certainement trouver une fonction de coût modélisant ce processus, il nous semble cependant plus pertinent et porteur de sens d'aborder ce phénomène sous l'angle d'une adaptation dynamique des mouvements en fonction de la manière dont se passe l'interaction.

Afin de compléter cette étude, nous allons maintenant nous intéresser aux structures cérébrales et à leurs rôles possibles pour le contrôle moteur.

1.2 Neurobiologie du contrôle moteur

Nous allons maintenant aborder les structures cérébrales et leur rôle actuellement reconnu dans le contrôle moteur et les apprentissages participant à ce contrôle. Nous verrons que si les aspects d'optimisation ont une réalité biologique, les processus liés au contrôle moteur réalisés par le cerveau ne peuvent s'y limiter. Nous allons nous attarder sur les aspects de contrôle haut niveau des mouvements. Ce contrôle est effectué dans le cerveau au niveau de plusieurs structures dont je vais détailler ici les caractéristiques (Figure 1.1). Les 3 principales structures sont le cervelet, le striatum et l'hippocampe. Elles s'articulent autour du cortex moteur que nous aborderons également.

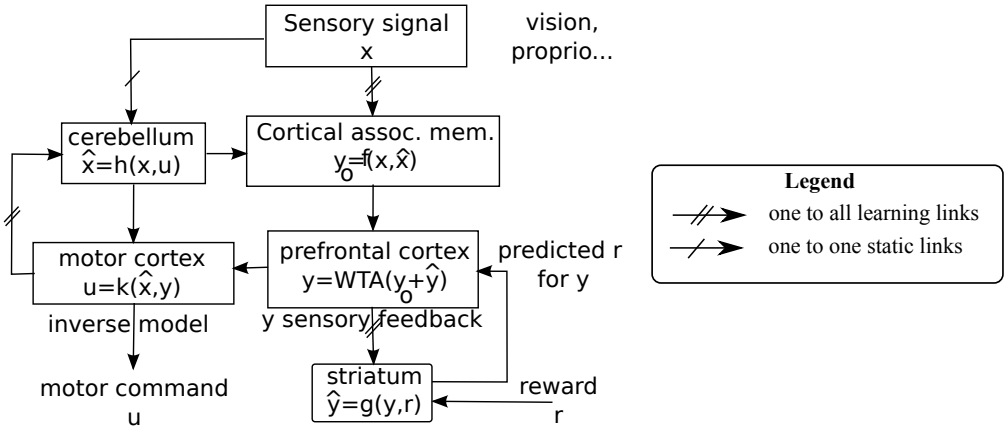


FIGURE 1.1 – Modèle computationnel du cerveau humain adapté à partir de [Bhushan and Shadmehr, 1999]

1.2.1 Cortex pariétal, cortex moteur et cortex prémoteur

Le cortex pariétal, le cortex moteur et le cortex prémoteur sont le siège d'apprentissages associatifs entre les différentes sensations et les commandes motrices. Ils permettent le pavage de l'espace proprioceptif lors de phase de babillage et le raffinement de ces états lors du fonctionnement normal. Le cortex pariétal rassemble et fusionne les informations sensorielles notamment visuelles. Ces informations sont combinées avec le retour proprioceptif au niveau du cortex prémoteur. Le cortex moteur s'appuie sur le retour proprioceptif et les activités présentes au niveau du cortex prémoteur pour générer les signaux déclenchant et contrôlant le mouvement volontaire des muscles. Idéalement, la génération des commandes motrices pourrait aussi être réalisée quand les retours sensoriels externes sont absents, juste en utilisant les prédictions

du cervelet et du striatum. Les commandes motrices calculées par le cortex moteur sont ensuite converties en activation des muscles agoniste/antagoniste. Les travaux de Georgopoulos et collègues [Georgopoulos et al., 1986] ont montré l'existence chez le singe de neurones dans le cortex moteur qui répondent de manière sélective à l'orientation d'un mouvement d'atteinte réalisé. Cet encodage en population du mouvement est à la base des modèles dynamiques décrits à la section 1.3.

1.2.2 Cervelet

Le cervelet est la structure cérébrale la plus imposante du point de vue du nombre de neurones. On estime que bien qu'il n'occupe que 10% du volume du cerveau il contient plus de neurones que le reste du cerveau [Shepherd, 2003]. Ce simple fait a suffi pour susciter la curiosité. Nombre d'études tant du côté des neurobiologistes que des modélisateurs ont vu le jour durant les 50 dernières années et les connaissances accumulées sur le sujet sont conséquentes. Cependant il reste beaucoup de zones d'ombre quant à la compréhension du fonctionnement de cette structure.

Le cervelet est situé au-dessous de la masse principale du cerveau et juste derrière le Pons. Cette dernière structure est la seule connexion du cervelet avec le reste du cerveau et toutes les informations transitent par ce canal. Au niveau cellulaire la structure du cervelet est étonnamment régulière. Elle est composée de structures répétitives constitués principalement de deux types de cellules : les cellules de Purkinje et les cellules Granulaires. Les cellules de Purkinje sont caractérisées par leur arborescence dendritique impressionnant qui se ramifie jusqu'à 200000 points terminaux [Shepherd, 2003]. Les cellules Granulaires sont, quant à elles, les plus petites et les plus nombreuses dans le cerveau humain. Les cellules formant le cervelet sont reliées entre elle par deux types de fibres : les fibres moussues et les fibres montantes.

Bien que les connaissances sur le cervelet du point de vu de la biologie soient avancées, son ou ses rôles précis dans le fonctionnement cérébral reste flou. On sait depuis longtemps qu'il est fortement impliqué dans l'apprentissage de prédictions motrices [Holmes, 1939]. Depuis il a été impliqué dans l'apprentissage et la reconnaissance de séquences et de motifs [Albus et al., 1971; Gilbert, 1974], et l'apprentissage de conditionnements moteurs [Clark et al., 1984; McCormick and Thompson, 1984]. Des recherches plus récentes orientées dans le domaine du contrôle moteur montrent que le cervelet est responsable de l'apprentissage des modèles directes et inverses du système moteur [Wolpert et al., 1998; Kawato et al., 1987]. Toutes ces implications expliquent la nécessité d'un si grand nombre de neurones dans le cervelet en comparaison avec les autres structures du cerveau.

La modélisation du fonctionnement du cervelet reste un défi. Plusieurs modèles se sont succédés en s'appuyant sur les travaux d'Eccles [Eccles et al., 1967]. Le modèle de [Marr, 1969] associe l'apprentissage de compétences motrices au cervelet. Selon sa théorie, le télencéphale (structures corticales et sous corticales parmi lesquelles les ganglions de la base et le cortex limbique) organise le mouvement en venant activer les muscles via le cervelet. Les activités transitant par le cervelet permettent l'apprentissage de réflexes conditionnés. Par la suite, le retour sensoriel pourrait déclencher une réponse immédiate au niveau du cervelet sans nécessité un traitement au niveau des structures corticales. Cependant, ces réflexes seraient conditionnés à des contextes au niveau cortical. Le télencéphale pourrait ainsi utiliser les contextes pour "programmer" les routines motrices exécutées par le cervelet. La théorie de [Albus et al., 1971] se

focalise sur l'apprentissage et la reproduction de séquences motrices dans le cervelet ainsi que sur la modulation des commandes musculaires pour obtenir un contrôle précis des mouvements. Son apport est aussi de montrer la correspondance entre certains apprentissages du cervelet et les apprentissages d'un réseau de type perceptron [Rosenblatt, 1958] dans lequel les entrées seraient recodées pour faciliter l'apprentissage de la réponse réflexe. En 1975, Albus a proposé le Cerebellar Model Articulation Controller (CMAC) [Albus, 1975] qui s'appuie sur sa théorie du recodage associé à un apprentissage de type perceptron pour apprendre différents contrôleurs. Le principe du recodage consiste à projeter sur une base de dimension plus grande les données afin de faciliter et d'accélérer les apprentissages en évitant les interférences. Le modèle CMAC a été utilisé pour implémenter un contrôleur adaptatif temps réel pour différentes applications robotiques [Miller, W., 1987]. Des recherches plus récentes se sont penchées sur la modélisation du cervelet comme constructeur du modèle direct et inverse pour les systèmes robotiques [Kawato et al., 1987; Wolpert et al., 1998]. On notera également que plusieurs modèles applicatifs dans le domaine de la robotique ont été proposés. Les auteurs de [Carrillo et al., 2008] présentent un modèle à spike assez détaillé qui permet de contrôler un bras robotique en permettant une adaptation dynamique lorsque le contexte change (friction, charge). Ce modèle présente des résultats impressionnant en termes de robotique autonome. Il est cependant impossible de passer à l'échelle d'un robot à 6 DOF comme dans notre cas.

1.2.3 Ganglions de la base et boucles cortico-striatales

Les ganglions de la base sont composés de plusieurs structures dont le striatum [Beiser et al., 1997]. Le nom de striatum n'est apparu qu'en 1941 pour simplifier la nomenclature. Il est constitué du noyau caudé et du putamen séparés par la capsule interne. Bien que la complexité de sa structure soit toujours un frein pour comprendre toutes les facettes de ses fonctionnalités, les recherches ont cependant montré que le striatum est impliqué dans le contrôle moteur [Aosaki et al., 1995] et qu'il joue un rôle de modulateur [Alexander et al., 1986]. Ce rôle de modulateur s'appuie sur le fait que les ganglions de la base sont impliqués dans des tâches d'apprentissage par renforcement [Schultz, 2000; Doya, 1999] au travers des décharges de dopamine permettant le renforcement des connections synaptiques au niveau du striatum. Cependant le rôle des ganglions de la base ne se limite pas à cet aspect d'apprentissage. Les boucles cortico-basales sont structurées en voies parallèles mises en compétition permettant d'avoir un processus de sélection assimilable à de la sélection d'action [Redgrave et al., 1999; Gurney et al., 2001; Dollé et al., 2010]. Dans notre modèle, le rôle des ganglions de la base sera réduit à cette sélection et les actions à réaliser seront sélectionnées en fonction des différentes activités transmises par l'hippocampe et le cortex préfrontal.

1.2.4 Hippocampe

L'hippocampe tient son nom de sa forme qui rappelle fortement celle de l'animal marin. Il se situe dans le lobe temporal médian et appartient au système limbique. Il se compose de trois sous-structures : le subiculum, la corne d'Ammon (composée des aires CA1, CA2 et CA3) et le Gyrus Dentelé (DG). Cependant nous n'aborderons pas toutes ses sous-structures en détails. L'hippocampe semble avoir trois fonctions : il participe à la localisation spatiale (cellules de lieu dans l'hippocampe du rat [O'Keefe and Nadel, 1979]), il est impliqué dans la mémoire épisodique chez l'humain et l'animal [Eichenbaum et al., 1994] et il semble capable de détecter

la nouveauté grâce à sa capacité à prédire des évènements multimodaux[Banquet et al., 1997]. Certaines observations [Albouy et al., 2008] laissent également penser que l'hippocampe et le striatum fonctionnent ensemble sur les tâche d'apprentissage de séquences de mouvements et jouent tous les deux un rôle important. [Berger and Thompson, 1978] ont montré que l'hippocampe participait à l'acquisition de conditionnements simples notamment grâce à ses relations avec le septum [Hasselmo and Schnell, 1994] qui module les apprentissages corticaux et cérébelleux. Dans la suite, nous utiliserons le modèle de [Banquet et al., 1997; Gaussier et al., 1998] inspiré des analogies entre le cervelet et l'hippocampe. Nous considérerons que l'hippocampe et le cervelet ont la capacité à prédire le délai entre deux événements simples grâce aux bases de temps qui émergent des interactions entre cellules granulaires et cellules moussues (voir aussi les modèles de [Bullock et al., 1994; Reiss and Taylor, 1991]). Les cellules de Purkinje dans le cervelet ou les cellules pyramidales dans CA3 apprendraient à prédire par simple conditionnement l'événement suivant sur la base de la signature temporelle fournie par les cellules granulaires (ce modèle abstrait ne fera pas la différence entre le rôle inhibiteur des cellules de Purkinje et le rôle excitateur des cellules pyramidales qui doivent cependant traduire des différences comportementales importantes). Pour des raisons de simplicité, nous resterons au niveau d'un modèle abstrait qui sera utilisé pour l'apprentissage du timing entre événements simples ou pour la détection de nouveauté.

1.2.5 Cortex préfrontal

Le cortex préfrontal est connu pour abriter des capacités cognitives dites supérieures. On distingue classiquement trois régions préfrontales principales : le cortex cingulaire antérieur, le cortex orbitofrontal et le cortex préfrontal dorso-latéral [Goldman-Rakic, 1987; Fuster, 2008].

Le cortex préfrontal dorso-latéral a été associé aux fonctions cognitives mises en jeu dans le contrôle exécutif[3] telles que l'attention exécutive [Miller et al., 2002], la mémoire de travail [Baddeley and Hitch, 1983], la planification [Banquet et al., 1997; Hasselmo, 2005], la prise de décision. Le cortex orbitofrontal jouerait un rôle dans les processus affectifs et motivationnels [Bechara et al., 2000] comme l'inhibition, le codage de la valeur motivationnelle d'un stimulus, la prise de décision et le contrôle de l'action basés sur la récompense [Landmann, 2007], et le contrôle de l'humeur. Enfin, le cortex cingulaire antérieur serait impliqué dans le contrôle des fonctions autonomes, l'initiation de la réponse, l'intention, l'inhibition, le traitement du conflit ou de l'erreur et l'allocation des ressources cognitives [Bush et al., 2000; Botvinick et al., 2001].

Bien que nous ayons présenté les principales régions du cortex préfrontal pour illustrer la diversité des fonctions cognitives réalisées par celui-ci, nous ne différencierons pas ces régions dans nos modèles. Cette décision est renforcée par notre focalisation sur la question du contrôle moteur. Nous considérerons principalement le cortex préfrontal pour son rôle dans le contrôle exécutif en lien avec des aspects motivationnels permettant de définir des comportements de planification (Sec. 1.5.1).

3. le contrôle de l'exécution des actions

1.2.6 Bilan

Afin de rendre compte des capacités de contrôle moteur de l'humain, il faut considérer le cerveau dans son ensemble. Par exemple, Bhushan et Shadmehr [Bhushan and Shadmehr, 1999] soulignent la nécessité du couplage modèle inverse/modèle direct pour un système qui doit rendre compte des observations faites chez l'humain. D'après leurs recherches, un modèle inverse seul ou un modèle direct seul ne suffit pas à reproduire les mouvements humains, même en permettant une adaptation en ligne du modèle utilisé. Ils proposent un modèle global du cerveau attribuant différents rôles complémentaires aux différentes structures cérébrales (Figure 1.1). On aboutit à l'idée que les structures cérébrales correspondent principalement à des mécanismes d'apprentissage [Doya, 1999] mis à disposition de l'ensemble du cerveau afin de permettre la réalisation des comportements. La théorie de la réutilisation neuronale (*neural reuse*) proposée par [Anderson, 2010] insiste sur le fait que les composantes du cerveaux se caractérisent par leur capacité à réutiliser les signaux disponibles, y compris ceux générés par les autres composantes. Les comportements résultent donc des échanges d'information (interactions) entre les différentes régions cérébrales, qui ne peuvent être limitées à une tâche particulière. Ainsi, il faudrait aborder le problème du contrôle moteur à travers un modèle qui reprenne les multiples régions du cerveau en tenant bien compte des échanges possibles entre les différents niveaux du modèle.

Afin d'étudier le lien entre le cerveau et les comportements, Eliasmith et al. [2012] proposent le modèle SPAUN utilisant quelques millions de neurones à décharges. Ce modèle a permis qu'un agent simulé réaliser différents comportements cognitifs (reconnaissance de chiffres, reproduction/complétion de l'écriture d'un chiffre, etc.). L'objectif principal des auteurs est de proposer un ensemble unifiés de mécanismes capable de réaliser les différentes tâches considérées. De ce fait, ils ne cherchent pas à expliquer la mise en place de ces différents comportement suivant une approche épigénétique. De manière assez similaire, O'Reilly propose une architecture cognitive, c'est à dire permettant d'obtenir différents comportements cognitifs humains, en s'appuyant sur différents mécanismes ou principes biologiques O'Reilly [1998]. Le modèle computationnel bio-inspiré appelé LEABRA O'Reilly [1997]; O'Reilly et al. [2013] et ses principes sont basés sur l'étude de nombreuses de données biologiques. Ce modèle a permis d'obtenir différents comportements de reconnaissance d'objet et d'attention visuelle avec une implémentation du modèle sur un agent simulé. On notera que leurs travaux s'appuient initialement sur une validation en terme de correspondance avec les données biologiques notamment au niveau des schémas d'activation dans les différentes structures impliquées lors de la réalisation des comportements. Toutefois, la robotique incarnée fait partie de leurs perspectives en cours. Dans cette thèse, nous nous intéresserons aussi à la mise en place d'une architecture cognitive, cependant notre approche privilégiera la validation par les comportements obtenus, c'est pourquoi nous mettrons en avant l'implémentation sur robot réel pour observer la relation entre le modèle et les comportements du robot. D'autre part, nous nous intéresserons aussi la manière dont les comportements du robot vont pouvoir se développer en même temps que les capacités de l'architecture.

1.3 Approche dynamique du contrôle moteur

Les aspects de contrôle optimal semblent principalement localisés dans la boucle cortico-basale avec les aspects de renforcement et le cortex-préfrontal suivant la capacité à représenter les objectifs et les motivations permettant de choisir à haut niveau le comportement désiré. Cependant, la mise en place de l'encodage des trajectoires désirées ou des primitives de mouvements est difficile à ancrer parmi les différentes structures cérébrales. A plus bas niveau, les résultats de Georgopoulos et collègues [Georgopoulos et al., 1986] et les modèles de l'approche dynamique des comportements montrent qu'un minimum de mécanismes sensorimoteurs sont suffisants pour définir des comportements.

1.3.1 Comportements dynamiques et champs de neurones dynamiques

Les DNF (Dynamic Neural Fields) [Schoner, 1995] sont une solution intéressante pour modéliser le contrôle moteur et rendre compte d'un grand nombre de résultats en psychologie (stabilité et capacité de bifurcation de notre contrôle moteur). Les DNF servent de base à l'application de la théorie des systèmes dynamiques pour modéliser les processus développementaux sensorimoteurs chez le très jeune enfant [Thelen et al., 2001]. Dans un DNF, une variable comportementale θ peut être encodée par une population de neurones. Chaque neurone représente alors une valeur de la variable. L'activité u du champ de neurones est mise à jour selon l'équation d'Amari [Amari, 1977].

$$\tau \dot{u}(\theta, t) = -u(\theta, t) + \int w(\theta - \theta') f(u(\theta', t)) d\theta' + h + I(\theta, t) \tag{1.9}$$

avec τ la constante de temps du système, w un noyau d'interaction par exemple une différence de gaussiennes, f une fonction seuillage de type sigmoïde et h une constante déterminant une quantité globale d'inhibition ou d'excitation dans le système. I représente le potentiel d'entrée du système. La dérivée du signal d'activation du DNF peut fournir un profil de commande en vitesse. A chaque instant, la valeur de la variable comportementale permet de lire dans le champ dérivé la valeur de la commande en vitesse (voir Fig. 1.2 a et b). La dynamique des mouvements obtenus avec un DNF implique que les maxima locaux du champ neuronal sont des points attracteurs. Suivant la situation initiale, le système convergera vers l'attracteur le plus proche. Cette propriété de bifurcation permet de faire une sélection cohérente malgré la présence de multiples stimuli en entrée. De plus, la dynamique interne des neurones donne des capacités de mémoire qui permettent de filtrer les stimuli perceptifs non stables en induisant une hystérésis dans la sélection de l'attracteur suivi. L'approche des DNF insiste sur l'importance du couplage entre les signaux sensoriels activant un DNF et les actions générées par lui. Thelen et collègues [Thelen et al., 2001] ont appliqué cette approche au phénomène de persévérance des gestes d'atteinte chez les enfants entre 7 et 12 mois que Piaget avait initialement attribué à l'absence de la notion de persistance des objets chez les très jeunes enfants. L'expérience de A non B de Piaget décrit une situation où un enfant qui a réussi à découvrir un objet caché à un endroit "A" continuera d'essayer d'atteindre ce même endroit après avoir observé que l'objet avait été déplacé et caché à un autre endroit "B". Ce phénomène est traditionnellement considéré comme une étape clé dans l'apprentissage de la permanence des objet i.e. le fait que les objets continuent d'exister même quand ils ne sont plus visibles. Dans [Thelen et al., 2001], les auteurs ont montré que ce phénomène pouvait être expliqué sur la simple base du couplage dynamique

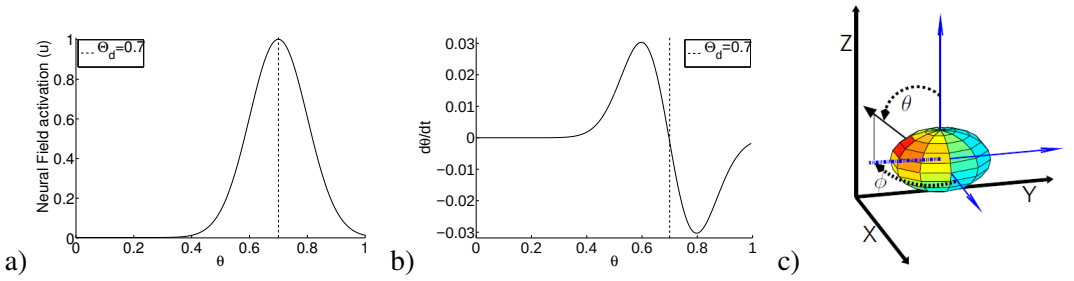


FIGURE 1.2 – Activation gaussienne d'un champ neuronal (a) et champ dérivé correspondant (b) pouvant fournir une commande en vitesse $\dot{\theta} = \frac{du}{d\theta}$. (c) Contrôle d'un bras à multiple degrés de liberté. La position de l'extrémité du bras (sphère) est donnée dans un référentiel absolu fixé sur la surface de travail (z=0). La direction instantanée du mouvement de l'extrémité est représentée par deux angles l'un lié à l'axe vertical du référentiel absolu (élévation θ) et l'autre à l'axe horizontal x (azimut ϕ). Le champ de neurones codant la direction de mouvement est alors représenté par une sphère. Les différentes couleurs indiquent l'activité des neurones du champ, le maximum (rouge) indique la direction privilégiée. Tiré de [Iossifidis and Schoner, 2004]

entre perception et action. A cause des propriétés de mémoire et d'hystérésis des DNF, le fait que l'enfant ait atteint plusieurs fois l'objet en "A" renforce l'attracteur moteur en "A" induisant le comportement d'atteinte erroné lorsque l'objet est déplacé en "B". Les processus dynamiques permettent d'expliquer ce phénomène sans faire intervenir de représentations mentales liées à la notion d'objet. Dans [Thelen et al., 2001], les DNF sont positionnés comme représentation motrice de l'action à réaliser. Ruffman, dans son commentaire [Ruffman, 2001] de [Thelen et al., 2001], questionne le choix de ce positionnement en remarquant que les mêmes propriétés de mémoire et d'hystérésis dans un processus attentionnel permettrait aussi d'expliquer le phénomène du "A non B". Par ailleurs, différents travaux [Johnson et al., 2009; Rougier, 2009; Fix et al., 2010] se sont effectivement intéressés aux DNF pour rendre compte de l'attention visuelle et de mémoire attentionnelle. Ainsi, les DNF pourraient être appliqués pour modéliser différents niveau de contrôle moteur et de traitement des informations sensorielles.

Les champs de neurones dynamiques ont été utilisés en robotique pour réaliser différents types de contrôle : en navigation [Schoner, 1995] mais aussi pour le contrôle d'un bras [Iossifidis and Schoner, 2004; Reimann et al., 2011; Gaussier et al., 1998; Andry et al., 2004]. Chaque contrôle implique de choisir soigneusement la variable comportementale utilisée. En navigation, ce choix est assez simple. Comme le robot se déplace dans un espace 2D sur le sol, la variable comportementale est l'angle donnant la direction du mouvement mesurée à partir d'un nord absolu. Dans le cas du contrôle d'un bras, le nombre de dimensions augmente et il devient impossible de projeter les actions sur une unique variable. Dans [Iossifidis and Schoner, 2004; Reimann et al., 2011], les variables comportementales sont deux angles (ϕ, θ) donnant l'orientation du mouvement dans l'espace cartésien (Fig. 1.2c). Même si le contrôle est efficace, il suppose de connaître a priori le modèle inverse permettant de passer de l'espace 3D à l'espace moteur articulaire. Une autre solution utilisée dans [Andry et al., 2004] est de réaliser le contrôle dans l'espace visuel. Le robot apprend des associations visuomotrices i.e. la correspondance entre une configuration motrice et la position visuelle correspondant à l'extrémité de son bras. A partir de ces associations et étant donné une entrée proprioceptive, le robot peut extraire la position visuelle de l'extrémité. La cible visuelle est représentée par une bulle d'activité dans

deux DNF à 1 dimension (un en x et l'autre en y). La commande en vitesse est extraite par lecture de la valeur de la dérivée du champ correspondant à l'état proprioceptif (cf [Gaussier et al., 1998; Andry, 2002; Andry et al., 2004] pour plus de détails). Ce contrôle suppose une transformation simple entre la commande en vitesse dans l'espace visuel à 2 dimensions et l'espace moteur. Si le bras robotique dispose de plusieurs degrés de liberté (ddl) redondants et, surtout, non alignés avec les projections visuelles, alors un apprentissage particulier du modèle inverse devient nécessaire.

Les propriétés des DNF (bifurcation, mémoire, hystérésis) sont intéressantes pour le contrôle moteur et les comportements dynamiques en général. Cependant, les DNF présentent différents inconvénients. La dynamique des attracteurs nécessite un compromis entre l'amplitude de la commande en vitesse et le rayon d'action de l'attracteur. De plus, les modifications de cette dynamique sont difficiles à réaliser en ligne. La commande en vitesse obtenue par dérivation du DNF est bornée par la largeur de la bulle d'activité gaussienne générée dans le DNF. L'amplitude de la commande en vitesse dépend aussi de la bulle d'activité. Afin de modifier la dynamique, il faudrait pouvoir apprendre et adapter en ligne le noyau d'interaction, i.e. les connections entre les différents neurones du DNF. Au chapitre 2, je proposerai un modèle qui n'utilisera pas de DNF pour le contrôleur moteur. Néanmoins, il sera intéressant d'étudier si les propriétés des DNF sont aussi présentes dans mon contrôleur. Il faut préciser que l'usage des DNF ne sera pas complètement écarté, en particulier pour les aspects attentionnels liés au traitement des informations visuelles.

1.3.2 Contrôle sensorimoteur et coordination sensorimotrice avec représentation spatiale interne

Suivant les découvertes de [Georgopoulos et al., 1986] sur l'existence d'un codage en population de la direction du mouvement, Bullock et Grossberg [BULLOCK and GROSSBERG, 1989] ont proposé les modèles complémentaires VITE (Vector Integration To End-point) et FLETE (Factorization of LEngth and TEnsion) pour décrire le contrôle moteur chez l'humain et expliquer certaines propriétés neuromusculaires. Le modèle VITE permet d'intégrer dynamiquement la position désirée PPV transmis au modèle FLETE qui se charge de réaliser le contrôle musculaire correspondant à cette commande en position.

Le modèle VITE s'appuie sur une position cible TPV qui est comparée avec la commande en position actuelle PPV pour obtenir un vecteur différence DV. Ce vecteur DV est multiplié par un signal GO avant d'être intégré dans PPV. Le signal GO définit l'énergie avec laquelle le mouvement est réalisé. Selon sa valeur, le mouvement sera réalisé plus ou moins rapidement.

Le signal PPV de commande en position est reçu en entrée du système FLETE qui modélise la dynamique musculaire. Suivant une factorisation longueur-tension, FLETE permet de gérer aussi certaines propriétés du mouvement réalisé. Le modèle FLETE est détaillé en terme d'activation des neurones impliqués dans le contrôle musculaire et permet d'interpréter différentes propriétés anatomiques du contrôle neuromusculaire (e.g. rôle des neurones de Renshaw, influence du principe de la taille dans le recrutement des fibres musculaires).

Le champ de neurones PPV a un rôle central et critique dans l'architecture. Il rend la trajectoire explicite et permet de commander le contrôle musculaire à bas niveau. Lors des mouvements passifs du bras, PPV est mis à jour via le retour proprioceptif (Figure 1.3b). Cependant, lors d'un contrôle volontaire des mouvements, ce retour proprioceptif est bloqué et la mise à jour ne

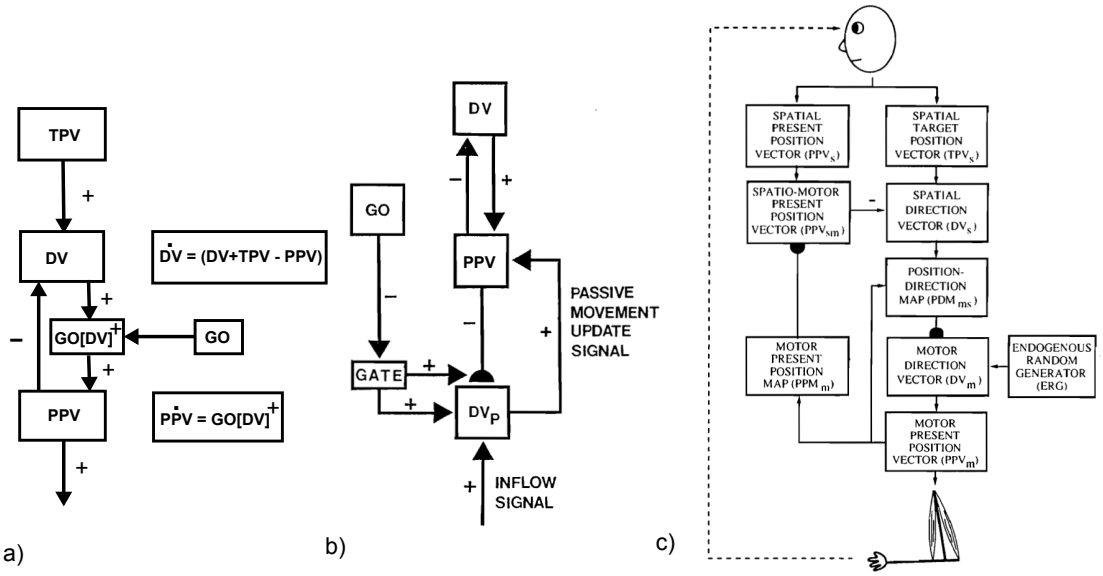


FIGURE 1.3 – *a) et b) :* Modèle VITE. Le modèle VITE contrôle la position du bras PPV en intégrant le vecteur direction DV calculé à partir de la différence entre la position cible TPV et la position actuelle PPV (a). Le signal GO module la vitesse avec laquelle le mouvement est réalisé. Le signal PPV peut être mise à jour lors des mouvements passifs grâce au retour proprioceptif (inflow signal) (b). *c) :* Modèle DIRECT. Le modèle s'appuie sur le modèle VITE. On intègre toujours un vecteur direction DVm sur la position désirée PPVm. Cependant, ce vecteur DVm est obtenu par la transformation depuis un espace interne 3D vers l'espace articulaire d'un vecteur spatial DVs. Ce vecteur DVs résulte de la comparaison de la position cible et de la position actuelle dans l'espace interne 3D.

peut se faire que par l'intégration de DV. Une gestion particulière de la fusion des informations proprioceptive et des mouvements désiré devrait être possible. Durant un mouvement volontaire, le retour sensoriel n'est pas pris en compte ce qui réduit les capacités d'adaptation du modèle lorsque PPV s'écarte de la réalité.

Hersch et Billard [Hersch and Billard, 2006] ont proposé un modèle de contrôle sensorimoteur capable de produire dans l'espace cartésien des mouvements orientés vers un but en évitant les singularités liées aux limites articulaires. Leur modèle utilise deux contrôleurs VITE en parallèle définissant deux types de contraintes sur le mouvement : une contrainte en position dans l'espace 3D cartésien et une contrainte en position articulaire. Les auteurs ont extrait du modèle VITE une écriture des équations permettant de rattacher chacune de ces contraintes à une forme de contrôle en impédance (sec. 1.1.2). Chaque contrôleur prédit une position suivante. Le contrôle final résulte alors d'une optimisation de l'écart entre les positions désirées (cartésienne et articulaire) et la position effectivement commandée selon une pondération de chaque contribution. Un modèle cinématique assure la cohérence entre la commande en position générée et les retours intégrés dans les deux contrôleurs VITE. La pondération en ligne des deux contraintes permet d'éviter les configurations singulières. Le contrôleur articulaire génère naturellement des trajectoires courbes qui permettent d'éviter les limites articulaires. Il suffit donc d'adapter la pondération pour privilégier la position prédite par ce contrôleur lorsque le bras s'approche trop de ses limites articulaires. Lorsque le bras est loin de ses limites, la contrainte en position dans l'espace cartésien peut être privilégiée permettant de générer un mouvement relativement

rectiligne dans cet espace.

Le modèle DIRECT (DIrection-to-Rotation Effector Control) [Bullock et al., 1993] améliore le modèle VITE en prenant en compte l'apprentissage de la coordination sensorimotrice. Le bras est contrôlé grâce aux informations visuelles et proprioceptives (Figure 1.3c). Ce modèle s'appuie sur une représentation spatiale 3D interne dans laquelle seront projetés tous les retours sensoriels (position visuelle de la cible et position visuelle de la main du robot, proprioception du bras robotique). La comparaison entre la position cible et la position actuelle dans l'espace 3D permet d'extraire la direction du mouvement. Cette direction est alors transformée, en tenant compte de la posture, en une commande motrice qui sera intégrée pour obtenir la position angulaire commandée (suivant le principe du modèle VITE). Dans le modèle DIRECT, les transformations de la vision ou de la proprioception vers l'espace 3D interne sont apprises. De même, le modèle inverse associant les combinaisons de postures et de directions de mouvement dans l'espace 3D avec des rotations articulaires est aussi appris. Grâce à ce modèle, le robot peut exhiber de nombreuses propriétés intéressantes pour le contrôle : le robot peut atteindre une position visuelle, contrôler son bras juste à partir de sa proprioception, adapter le contrôle lorsqu'un outil prolonge la main, réaliser le contrôle malgré des distorsions sur la vision...Une extension du modèle appelé DIRECT-ROAD [Srinivasa et al., 2012] propose d'une part d'améliorer les capacités d'apprentissage en utilisant le principe Fuzzy ARTMAP (développé par [Carpenter et al., 1992]) mais surtout permet d'éviter des obstacles lors des déplacements. L'utilisation d'un espace 3D pour représenter la direction du mouvement et un contrôle dynamique du comportement se retrouvent aussi dans [Iossifidis and Schoner, 2006]. On notera cependant que contrairement au modèle de [Iossifidis and Schoner, 2006], le modèle DIRECT-ROAD décrit comment apprendre la transformation entre la commande dans l'espace 3D et la commande en rotation ainsi que comment apprendre les transformations entre les espaces sensoriels et la représentation interne 3D.
Le contrôle DIRECT permet de construire la trajectoire au fur et à mesure des déplacements et permet de multiples propriétés d'adaptation du mouvement gratuites (sans réapprentissage). Cependant, ce modèle présente des limites selon l'approche développementale : il est difficile d'accepter la nécessité d'une représentation interne basée sur un espace 3D explicite. En effet, dans ce modèle, le contrôle moteur repose sur la comparaison des positions dans cet espace interne tant pour l'apprentissage du modèle générant les commandes en rotation que pour le contrôle par la suite. Les transformations des espaces sensoriels vers la représentation interne doivent donc être suffisamment bien calibrées. D'autre part, la construction de cette représentation interne 3D s'appuie uniquement sur la transformation de l'information visuelle stéréoscopique vers la représentation interne 3D centrée sur le corps. Différents travaux [Grossberg et al., 1993; Greve et al., 1993; Guenther et al., 1993] expliquent comment ce prérequis peut être appris de manière autonome. Cependant, le corollaire de ce modèle basé sur la vision est que les gens borgnes ou aveugles de naissance ne devraient pas pouvoir contrôler leur mouvement. Or, ces gens sont parfaitement capables d'agir sur leur environnement. Si l'existence d'une représentation interne 3D me paraît acceptable, il me semble déjà qu'elle devrait se construire à partir des mouvements et des déplacements réalisés plutôt qu'à partir du regard. Ensuite, il me semble plus probable que cette représentation interne soit construite progressivement en même temps que le contrôle moteur est appris. Elle ne peut être considérée comme un prérequis nécessaire pour le contrôle. Ayant écarté la nécessité d'une telle représentation 3D pour le contrôle moteur, nous avons pu utiliser dans nos expérience un robot minimal avec une vision monoculaire. Nous

savons qu'à terme la stéréoscopie sera nécessaire pour réaliser des comportements plus complexes cependant avec ce dispositif robotique minimal, le développement de comportements sensorimoteurs intéressants est déjà possible.

1.3.3 Contrôle par sélection d'attracteurs - principe "Yuragi"

Les auteurs de [Fukuyori et al., 2008; Nurzaman et al., 2009; Sugahara et al., 2010] ont proposé un modèle de sélection d'attracteurs basé sur des fluctuations biologiques ("yuragi" en japonais) et s'appuyant sur l'équation de Langevin :

$$\Lambda \cdot \dot{\mathbf{x}} = \xi \cdot f(\mathbf{x}) + \eta$$

où Λ est une constante de temps, le vecteur $\mathbf{x}$ décrit l'état du système et la fonction f génère un bassin d'attraction résultant de la combinaison des attracteurs connus. Un signal de renforcement sur les performances actuelles du mouvement vient moduler le coefficient ξ pondérant la fonction d'attraction face à un terme d'exploration aléatoire η. Grâce à cette pondération variable de ξ face à η, le système peut alterner entre l'exploration aléatoire de l'espace moteur et l'exploitation d'un attracteur particulier (Fig. 1.4). Cette approche est particulièrement intéressante pour les capacités d'adaptation du comportement moteur qu'elle offre. En effet, un apprentissage frustre peut être complété par ce mécanisme afin d'obtenir les performances souhaitées. Par exemple, le contrôle moteur permet de rejoindre des parties de l'espace où aucun attracteur n'a été appris auparavant et ainsi découvrir de nouveaux mouvements. Suivant un cercle vertueux, le système peut s'appuyer sur les états explorés pour raffiner son apprentissage. Ainsi, dans les expériences réalisées [Fukuyori et al., 2008], les attracteurs peuvent même être positionnés aléatoirement au départ. L'exploration et l'apprentissage combinés permettent de rejoindre la cible désirée et de déplacer les attracteurs afin de pouvoir atteindre plus facilement cette cible. Un contrôleur "Yuragi" présente des similarités avec un apprentissage par renforcement. Il a par exemple été utilisé en ce sens lors d'une expérience sur l'interaction sociale dans laquelle un expérimentateur guide les gestes d'un robot par des renforcements positifs ou négatifs (expressions faciales) [Boucenna et al., 2010b].

D'autres travaux se sont intéressés à la possibilité d'alterner exploration et exploitation des attracteurs. Par exemple, Khamassi et collègues [Khamassi et al., 2010] proposent un modèle inspiré du cortex préfrontal et de la boucle cortico-striatale pour expliquer la modulation des comportements d'exploitation et d'exploration dans des tâches d'apprentissage par récompense. Ceci permet de mieux modéliser les stratégies d'action suivies par un singe à la recherche des bonnes actions pour obtenir un récompense.

Le principe du "Yuragi" met en avant la possibilité de réaliser dynamiquement un comportement désiré en s'appuyant sur les attracteurs déjà appris créant ainsi des attracteurs "virtuels". Dans mon modèle, au chapitre 2, je mettrai en oeuvre ce mécanisme et je discuterai les perspectives offertes par la combinaison d'attracteurs suivant le principe du "Yuragi".

1.4 Modèle PerAc : un modèle bio-inspiré pour la construction de comportements dynamiques

Parmi les modèles qui s'inscrivent dans l'approche dynamique des comportements, nous détaillerons plus particulièrement notre modèle Perception-Action (PerAc) [Gaussier and Zre-

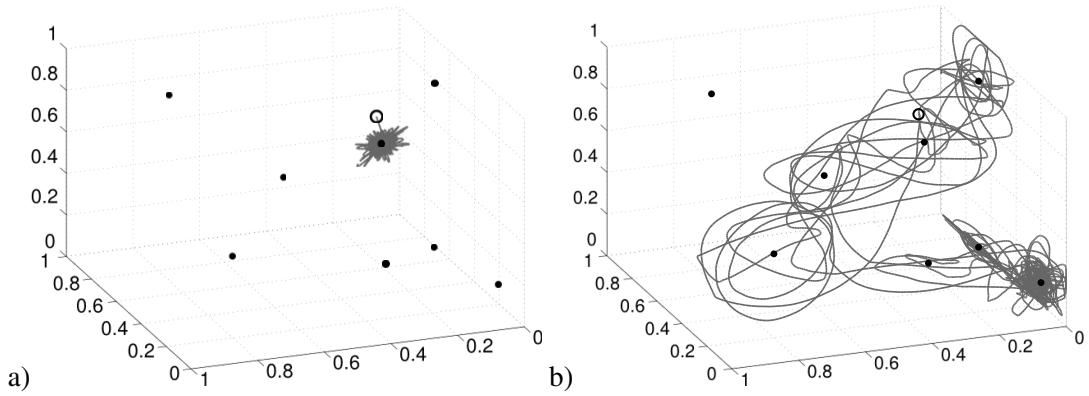


FIGURE 1.4 – Simulation des mouvements d'un bras à 3 degrés de liberté utilisant le principe du "Yuragi" (tiré de [de Rengervé et al., 2010a]). *a* : Si le retour est positif, la trajectoire converge vers l'attracteur le plus proche. *b* : Quand le retour est négatif, la force d'attraction des attracteurs diminue, et le bruit dévient prépondérant dans la commande. Le système explore alors l'espace moteur, passant d'un attracteur moteur à l'autre.

hen, 1995]. Le principe du modèle PerAc est exprimé par la possibilité de construire des comportements grâce à l'apprentissage d'associations sensorimotrices simples. Après avoir décrit ses mécanismes d'apprentissage, nous le comparerons avec LWPR [Vijayakumar et al., 2005] pour mieux situer son potentiel en terme d'apprentissage sensorimoteur. Nous présenterons alors l'application de PerAc à des problèmes de navigation suivi d'une comparaison entre PerAc et GMM/GMR [Calinon et al., 2009] sur l'apprentissage du suivi d'un chemin.

1.4.1 Les principes du modèle Perception-Action (PerAc)

Le modèle PerAc s'appuie principalement sur trois processus : catégorisation, compétition et conditionnement (Fig. 1.5). Des catégories s^X sont formées à partir de l'entrée sensorielle

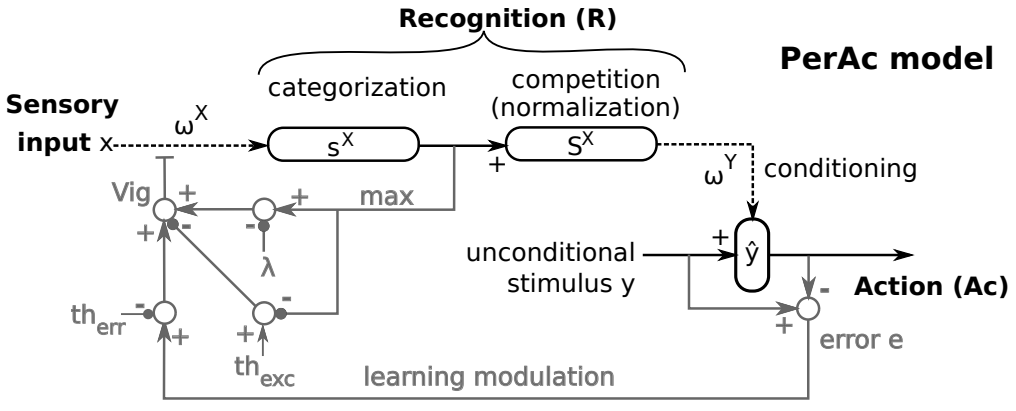


FIGURE 1.5 – Architecture PerAc (apprentissage d'associations simples et conditionnement) avec recrutement selon la reconnaissance de l'entrée (x) et l'erreur de prédiction sur la sortie (y). Les modules en noir correspondent aux processus d'apprentissage au centre de PerAc. Les modules en gris sont les traitements liés au calcul de la modulation de l'apprentissage. Cette architecture apprend à prédire une action Ac à partir de la reconnaissance R d'un état sensoriel.

x (eq. 1.10).

$$\begin{cases} s_i^X = e^{(-\frac{\sum_j (x_j - \omega_{ij}^X)^2}{2\beta^X})} \text{ et} & S_i^X = \frac{s_i^X}{\sum_i s_i^X}, \\ Vig = \mathcal{H}(\lambda - \max_i(s_i^X)) & +\mathcal{H}(e - th_{err}) \\ & -\mathcal{H}(\max_i(s_i^X) - th_{exc}), \\ \Delta\omega_{i'j}^X = Vig \cdot (x_j - \omega_{i'j}^X). \end{cases} \tag{1.10}$$

Les catégories s^X sont apprises dans un réseau de neurones inspiré par l'Adaptive Resonance Theory [Carpenter and Grossberg, 2002]. Le signal de recrutement Vig est actif quand le niveau de reconnaissance des catégories est inférieur à λ (terme de vigilance) mais aussi quand l'erreur e sur la sortie est inférieure à th_{err} (seuil d'erreur). Il s'agit de l'erreur au sens des moindres carrés entre la sortie prédite $\hat{y}$ et la sortie désirée y :

$$e = \sqrt{\sum (y_i - \hat{y}_i)^2} \tag{1.11}$$

S'il y a déjà une catégorie codant une situation similaire (activité de reconnaissance au dessus du seuil de reconnaissance th_{exc}), alors aucune nouvelle catégorie sera recrutée. Recruter selon le niveau d'erreur e permet d'améliorer la prédiction indépendamment d'une densité de recrutement fixée a priori par le seuil de vigilance λ. Le seuil de reconnaissance th_{exc} permet d'éviter de recruter trop de catégories pour des configurations similaires. Lors du recrutement, la configuration est encodée sur les poids synaptiques $\omega_{i'j}^X$ d'un neurone i', qui n'a pas encore été recruté pour représenter une catégorie. Les activités des neurones codant les catégories sont normalisées (S^X). D'autres [4] structures compétitives auraient pu être utilisées comme une compétition dure (WTA, Winner-Takes-All) ou une compétition souple (SWTA, Soft Winner-Takes-All). La normalisation présente l'avantage de mélanger les prédictions et donne donc des résultats un peu plus précis.

L'apprentissage associant la sortie y avec les catégories est basé sur l'algorithme des moindres carrés (LMS, Least Mean Square) de Widrow et Hoff [Widrow and Hoff, 1960]. Cet algorithme réalise un apprentissage par conditionnement via une simple descente de gradient minimisant l'erreur entre la sortie inconditionnelle y et la sortie conditionnelle $\hat{y}$ au sens des moindres carrées (eq. 1.12).

$$\begin{cases} \hat{y} = \sum_i \omega_i^Y \cdot S_i^X \\ \Delta\omega_i^Y = \epsilon \cdot S_i^X \cdot (y - \hat{y}) \end{cases} \tag{1.12}$$

Les poids synaptiques ω_i^Y sont adaptés suivant l'erreur de prédiction. Un facteur d'apprentissage ϵ relativement faible permet de moyenner les apprentissages.

L'algorithme PerAc peut rapidement recruter de nouvelles catégories quand il y a des erreurs sur la sortie prédite. La catégorisation de ces situations particulières permet de récupérer des réponses correctes rapidement ce qui est utile lors d'un apprentissage en ligne en situation d'interaction. L'algorithme peut aussi adapter la prédiction sans recrutement afin d'éviter d'utiliser de trop nombreuses catégories pour coder le même domaine de l'espace d'états. Le modèle PerAc prend tout son sens dans l'apprentissage associant un état sensoriel avec une action (ici,

4. On remarquera que si β^X tend vers 0, le comportement de l'ensemble catégorisation-normalisation se comportera comme une catégorisation combinée avec un WTA.

estimée en amont du stimulus inconditionnel). Dans ce cas, les couples sensorimoteurs appris pourront constituer un bassin d'attraction définissant un comportement.

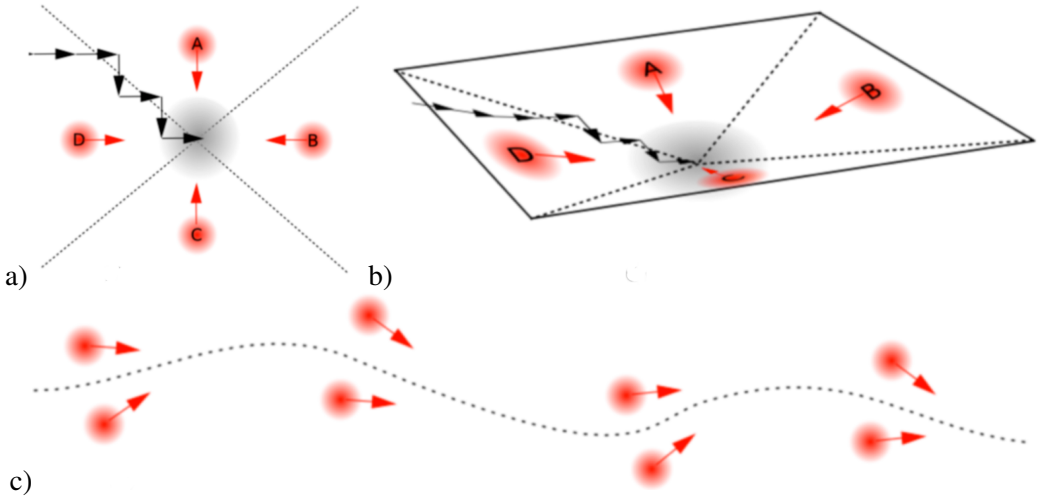


FIGURE 1.6 – Exemples de bassins d'attraction générés par des couples sensorimoteurs appris. Chaque flèche représente l'action associée à un état (cercle). *a,b)* Un bassin d'attraction basé sur quelques couples état/action suffit à générer le comportement pour rejoindre un état particulier de l'environnement. *c)* Le bassin d'attraction construit peut définir une trajectoire à suivre, par exemple une ronde.

La figure 1.6 illustre sur un espace d'état à 2 dimensions les comportements dynamiques qui peuvent être réalisés avec cette architecture sensorimotrice simple. Des couples sensorimoteurs sont appris (apprentissage de l'état et association de l'état avec l'action). A chaque instant, la reconnaissance R de l'état prédit l'action à réaliser Ac. Dans la figure 1.6 (b,c), un bassin d'attraction basé sur quelques couples lieu/action suffit à générer le comportement pour rejoindre une configuration particulière de l'espace d'état. Quelque soit son point de départ, le robot rejoindra la configuration apprise en généralisant son apprentissage sensorimoteur. Ces bassins d'attraction ne sont pas forcément des points fixes, on peut aussi créer des vallées correspondant à l'apprentissage de "routes" (figure 1.6 (d)). Par exemple, une telle architecture sensorimotrice utilisant la reconnaissance de cellules de lieu (R) associées avec l'orientation du mouvement (Ac) fournit un contrôle efficace pour la navigation et le suivi de chemins (sec. 1.4.3).

Dans l'approche PerAc, la perception est considérée comme le résultat de l'apprentissage sensation/action permettant un comportement global cohérent. Par exemple, par l'apprentissage d'associations sensorimotrices, un robot peut s'orienter vers un objet particulier donnant lieu à l'interprétation que le robot "perçoit" l'objet [Maillard et al., 2005]. Nous avons aussi proposé de définir mathématiquement la perception $\mathbf{Per}$ comme le champ potentiel associé à l'action $\mathbf{Ac}$ [Maillard et al., 2005]. Réciproquement, l'action $\mathbf{Ac}(\mathbf{p})$ en une position $\mathbf{p}$ de l'espace d'état est alors le gradient de la perception à cette même position.

$$\mathbf{Ac}(\mathbf{p}) = -m\nabla\mathbf{Per}(\mathbf{p}) \tag{1.13}$$

avec la constante m définissant l'inertie du système. L'intégration de l'action $\mathbf{Ac}$ réalisée sur un chemin $\mathcal{C}^p$ jusqu'à une position $\mathbf{p}$ de l'espace d'état donne l'équation suivante, à une constante

d'intégration près :

$$\mathbf{Per}(\mathbf{p}) = -\frac{1}{m}\int_{\mathcal{C}^p} \mathbf{Ac}(l)dl \tag{1.14}$$

Nous considérons que l'espace d'état est partitionné, la perception peut alors être représentée sous la forme d'une matrice **Per** associant un vecteur de perception pour chaque partition. L'équation (1.14) peut alors s'écrire :

$$\mathbf{Per} = -\frac{1}{m}\int_{\mathcal{C}^p} \mathbf{Ac}(l) \cdot \mathbf{Sen}^\top(l)dl \tag{1.15}$$

avec $\mathbf{Sen}(l)$ un vecteur dont chaque composante représente une partition et vaut 1 si l définit un point dans la partition et 0 sinon. Nous utiliserons cette définition de la perception au chapitre 2 dans la comparaison de notre modèle avec les DNF.

1.4.2 Comparaison entre PerAc et LWPR sur une approximation de fonction

Nous avons réalisé une comparaison entre l'algorithme LWPR (Locally Weighted Projection Regression) et le modèle PerAc dans un cas d'école : l'apprentissage d'une fonction non linéaire. Le modèle utilisé pour LWPR [Klanke et al., 2007; Vijayakumar et al., 2005] est celui fourni dans la librairie mise à disposition par les auteurs (`http://wcms.inf.ed.ac.uk/ipab/slmc/research/software-lwpr`). Le modèle PerAc [Gaussier and Zrehen, 1995] décrit à la section 1.4.1 a été appliqué à l'apprentissage de la fonction désirée.

Dans ce test, le facteur d'apprentissage ϵ est variable. L'équation d'apprentissage (1.12) du conditionnement est complétée par l'équation suivante :

$$\epsilon = \frac{max(\epsilon_m, Vig)}{||\ S^X\ ||} \tag{1.16}$$

Quand il y a recrutement ($Vig = 1$), la vitesse d'apprentissage ϵ est élevée pour permettre un apprentissage rapide de la configuration entrée-sortie. Puis, la vitesse d'apprentissage ϵ redevient faible ($\epsilon = \epsilon_m$) permettant aussi l'adaptation des configurations déjà apprises. La vitesse d'apprentissage ϵ dépend de la norme euclidienne de l'entrée S^X de la couche apprenant par conditionnement. Pour d'éviter des oscillations trop importantes sur les valeurs des poids de connexion durant l'apprentissage, l'apprentissage doit être plus rapide quand il n'y a qu'un seul neurone actif dans S^X et légèrement plus faible quand plusieurs neurones sont actifs en même temps. Grâce à la normalisation par la norme euclidienne $||\ S^X\ ||$ de l'entrée c'est le cas. Cette dépendance de la vitesse d'apprentissage ϵ au nombre de neurones actifs dans S^X permet un apprentissage moyen plus correct, car plus lent, lorsqu'il y a des interférences entre les catégories apprises.

La comparaison entre LWPR et PerAc a été réalisée en deux étapes. Dans un premier temps, les deux algorithmes LPWPR et PerAc doivent approximer une fonction modèle en forme de croix basée sur la combinaison de 3 gaussiennes (1.17). Ce test permet d'évaluer les modèles sur l'apprentissage d'une fonction non linéaire ayant pour entrée des vecteurs à 2 dimensions (x_1, x_2) et une sortie à une seule dimension (y).

$$\begin{cases} a = e^{-10 \cdot x_1^2}\ ,\ b = e^{-50 \cdot x_2^2}\ \text{ et }\ c = 1.25 \cdot e^{-5 \cdot (x_1^2 + x_2^2)} \\ y = max(a, b, c) \end{cases} \tag{1.17}$$

Les données d'apprentissage sont générées avec un bruit blanc compris entre $[-0.05; 0.05]$ sur la sortie y. L'apprentissage est réalisé durant 200 époques de 100 itérations. A la fin de chaque époque, les prédictions des deux modèles sont évaluées sur l'intervalle $[-1; 1]^2$ avec un pas de 0.05. Pour chaque couple (x_1, x_2), la sortie prédite $\hat{y}$ est comparée avec la valeur réelle y non bruitée.
Dans un deuxième temps, i.e. après l'instant $t = 20000$ itérations (200 époques), la fonction modèle est modifiée par l'ajout d'une bulle gaussienne étroite en $(0.6, 0.6)$ (eq. 1.18).

$$\begin{cases} a = e^{-10 \cdot x_1^2}\ ,\ b = e^{-50 \cdot x_2^2} \text{ et } c = 1.25 \cdot e^{-5 \cdot (x_1^2 + x_2^2)} \\ d = 1.25 \cdot e^{-75 * ((x_1 - 0.6)^2 + (x_2 - 0.6)^2)} \\ y = max(a, b, c, d) \end{cases} \tag{1.18}$$

Les deux algorithmes doivent alors adapter leur base de codage pour représenter la fonction modifiée. On rajoute alors 100 itérations avec évaluations durant lesquelles les modèles LWPR et PerAc vont apprendre la nouvelle fonction modèle.

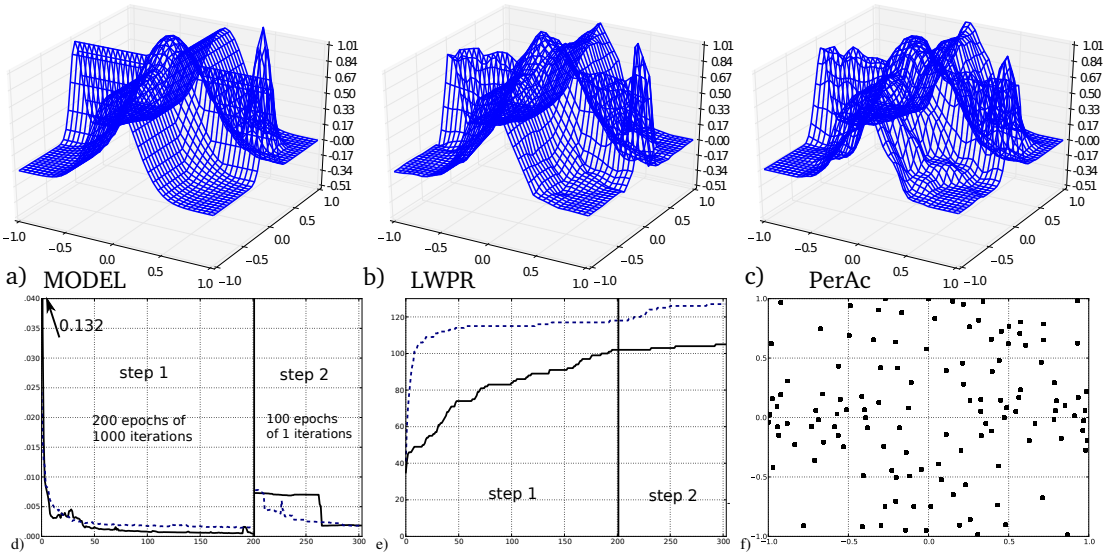


FIGURE 1.7 – Comparaison entre LWPR et PerAc sur un apprentissage simple. a) Fonction initiale devant être apprise. b) reproduction donnée par LWPR. c) Reproduction obtenue par PerAc. e) Erreur carrée moyenne à la fin de chaque époque d'apprentissage pour LWPR (trait plein) et PerAc (pointillé) (voir texte plus plus de détails). d) Évolution du nombre de catégories encodées pour LWPR (trait plein) et pour PerAc (trait pointillé). f) position des catégories apprises par PerAc. PerAc a besoin d'un nombre plus grand de catégories que LWPR pour des performances équivalentes. Les catégories sont recrutées là où il y a des variations importantes sur la sortie contrairement à LWPR où la répartition des catégories est plus uniforme.

A la fin de la première étape (après 20000 itérations d'apprentissage), les deux modèles ont des performances équivalentes (Figure 1.7) : LWPR a des performances légèrement meilleures que PerAc avec des erreurs carrées moyennes (mse, mean square errors) respectivement égales à $4.94e^{-4}$ et $1.5e^{-3}$. Les fonctions prédites sont qualitativement similaires à la fonction modèle, avec un résultat plus lissé pour LWPR. Les modèles LWPR et PerAc ont recruté un nombre comparable de champs récepteurs (catégories) : respectivement 102 et 118. Les paramètres des deux

	modèle LWPR	modèle PerAc
Principe	Régression sur des modèles linéaires paramétriques associés à des champs récepteurs locaux, projection des entrées pour réduire la dimension lors de l'apprentissage	Catégorisation de l'entrée avec recrutement sur la base de l'activité de reconnaissance versus un seuil de vigilance et des erreurs de prédiction importantes, et association par conditionnement
Apprentissage	Recrutement de champs récepteur, adaptation des paramètres des modèles linéaires (moindres carrés), adaptation de la taille et de la forme des champs récepteurs, apprentissage de la matrice de projection	Recrutement des catégories sur les entrées, associations avec la sortie basée sur un conditionnement simple (LMS)
Avantages	– apprentissage incrémental, local avec peu d'interférences entre les données apprises – *modèle utilisant des projections apprises sur un espace de plus faible dimension pour éviter la malédiction de la dimension* – prédiction basée sur la régression de modèles locaux linéaires pour une meilleure interpolation	– Apprentissage en-ligne rapide en situation d'interaction – Adaptation des prototypes encodés par les catégories – Modèle simple avec peu de calculs pour l'apprentissage et l'encodage des prédictions.
Inconvénients	– besoin de suffisamment de données pour mettre à jour correctement à la fois les champs récepteurs et les modèles linéaires associés – mise à jour de multiples variables statistiques pour réaliser les différents apprentissages	– pas d'adaptation de la sélectivité des catégories – pas de contrainte de lissage lors de l'apprentissage des prédictions

TABLE 1.2 – Résumé de la comparaison entre les algorithmes LWPR et PerAc.

algorithmes étant différents, on les a réglés pour recruter approximativement le même nombre d'états. PerAc recrute notablement plus rapidement, suggérant une approximation de la forme de la fonction modèle construite plus rapidement. Cette approximation est aussi plus grossière étant donné l'absence d'outil statistique sophistiqué lors de l'apprentissage. Durant la deuxième phase, le modèle PerAc s'adapte plus rapidement et donc a de meilleures performances durant les 50 premières itérations. Ce résultat est cohérent avec l'utilisation du modèle PerAc pour les apprentissages en ligne et les situations où le robot doit rapidement montrer le comportement désiré avec un minimum de démonstrations (coûteuses en temps). Sur le long terme, LWPR comble son retard et obtient des performances égales voire meilleures. De plus, on notera que LWPR peut apprendre des projections permettant d'optimiser l'encodage lorsque les données s'y prêtent (dans des variétés de faible dimension). Les principaux éléments de comparaison sont résumés dans la table 1.2.

1.4.3 Illustration de PerAc sur la construction de comportements de navigation

Suivant l'approche PerAc, l'équipe neurocybernétique du laboratoire ETIS a développé un contrôleur pour robots mobiles qui permet d'associer des informations visuelles (un panorama de l'environnement) avec l'orientation du robot (utilisant une boussole représentant la direction du mouvement actuel). Ce contrôleur définit une boucle sensorimotrice basée sur un modèle biologique s'appuyant sur les propriétés spatiales de l'Hippocampe découvertes en 1979 [O'Keefe and Nadel, 1979].

Définition des Cellules de Lieu

Afin d'être capable de se localiser et de naviguer, le robot utilise la reconnaissance de cellules de lieu basées sur des indices visuels (cf. [Giovannangeli et al., 2006; Lagarde et al., 2010] pour plus de détails à propos de l'extraction des caractéristiques visuelles). Un lieu est défini par une constellation de caractéristiques visuelles (couples amers-azimuts) extraites d'un panorama (Figure 1.8) et compressées dans un code. Le système visuel extrait des vues locales centrées sur des points d'intérêt (reconnaissance d'amers) qui fournissent l'information du "quoi". Une boussole magnétique donne une information proprioceptive sur l'orientation du robot qui, combinée avec la localisation des amers dans le champ visuel, fournit l'information du "où". Chaque caractéristique visuelle résulte de la fusion des informations du "quoi" et du "où" (Figure 1.9). La fusion de l'information est réalisée dans un espace produit *i.e.* une matrice de neurones produit $m_k(t)$ qui représentent chacun une caractéristique visuelle. L'ensemble des caractéristiques visuelles reconnues sur un panorama donné définit un code $M(t)$ représentatif du lieu dans lequel se trouve le robot. Les activités des cellules de lieu sont construites comme le résultat du calcul de la distance entre le code lieu appris et le code lieu actuel. L'activité $PC_p(t)$ de la p^e cellule de lieu est :

$$PC_p(t) = \frac{1}{W_p}\left(\sum_{k=1}^{n_M} \omega_{kp}^{PC}(t) m_k(t)\right) \tag{1.19}$$

où $\omega_{kp}^{PC}(t)$ exprime le fait que le couple amer-azimut k (*i.e.* la k^e caractéristique visuelle d'activité $m_k(t)$) a été utilisé pour catégoriser la cellule de lieu p. Le nombre de couples utilisés

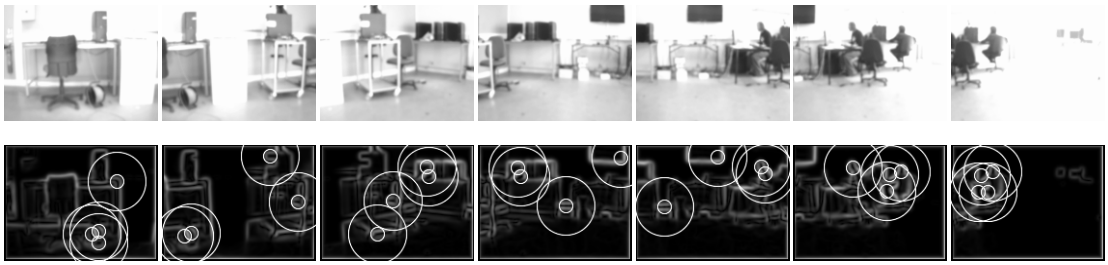


FIGURE 1.8 – Exemple de caractéristiques visuelles extraites sur un demi panorama visuel. Le système calcule un gradient sur chaque vue convolué ensuite avec une Différence de Gaussiennes (DOG, Difference of Gaussians). Les maxima locaux résultants pris par ordre d'intensité décroissant définissent la séquence d'exploration des points d'intérêt (points saillants). Ici, 4 points sont extraits à partir de chaque vue.

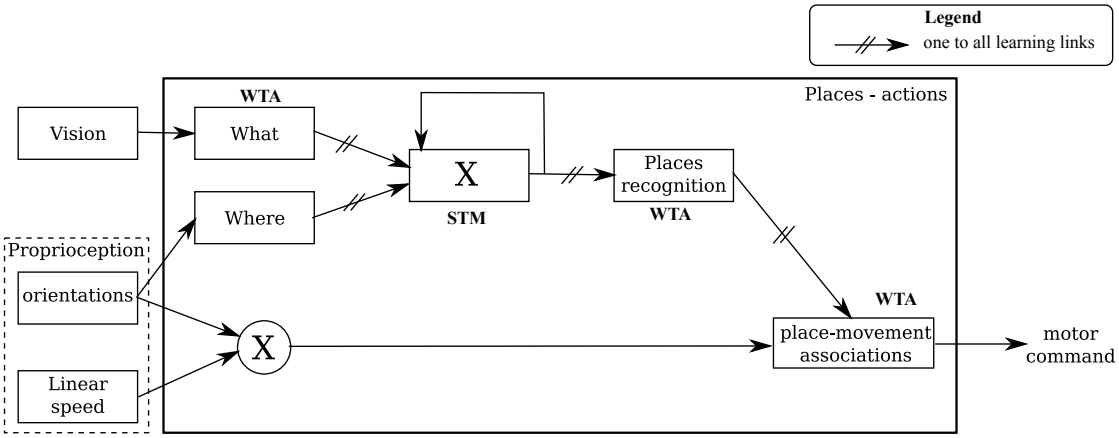


FIGURE 1.9 – Modèle pour l'apprentissage des associations lieu-action. Les informations de "quoi" et "où" sont extraites à partir de la vision et de la proprioception. Elles sont fusionnées et compressées dans des codes de lieu. Les activités des codes de lieux dépendent aussi d'une mémoire à court terme (STM, Short Term Memory) des précédents codes observés, permettant une reconnaissance des lieux avec une activité décroissante de manière monotone avec la distance au lieu appris. Ces codes de lieux sont associés avec l'information proprioceptive i.e. avec l'orientation du robot pour constituer des bassins d'attraction.

par la p^e cellule de lieu est donné par $W_p = \sum_{k=1}^{n_M} \omega_{kp}^{PC}$, avec n_M le nombre de neurones recrutés dans la matrice produit amer-azimut. Une cellule de lieu apprise dans le lieu A répond de manière maximale en A et crée un large champ décroissant autour de ce lieu. L'apprentissage de plusieurs régions de l'environnement et une compétition entre les cellules de lieu permet au système de réaliser une localisation basée sur la vision.

Associations lieu-mouvement pour construire des bassins d'attraction

Initialement, le robot suit une direction quelconque. Quand le robot prend une direction incorrecte, le professeur humain peut utiliser une laisse pour tirer le robot vers le chemin désiré. L'utilisateur modifie ainsi la dynamique du robot en forçant indirectement la commande motrice des roues à orienter le robot dans la direction désirée. Le changement de proprioception résultant (changement d'orientation) déclenche l'apprentissage conditionnel entre le mouvement réalisé (orientation et vitesse linéaire) et la localisation du robot i.e. les cellules de lieu calculées par le processus décrit ci-dessus. Les orientations et les vitesses sont discrétisées sous la forme de champs de neurones où chaque neurone correspond à une valeur particulière. En particulier, le codage en population utilisé pour contrôler l'orientation implémente un champ de neurones dynamiques (DNF, Dynamic Neural Field) ([Schoner, 1995], Sec. 1.3.1). Les activités du champs de neurones représentant l'orientation désirée suivent une bulle gaussienne centrée sur le neurone codant l'orientation désirée. La vitesse d'avancement linéaire est constante dans les implémentations les plus simples. Soit S_i l'activité d'un neurone codant une direction de mouvement (i.e. orientation ou vitesse selon le neurone) prédite par le modèle tandis que S_i^d est la direction de mouvement forcée par l'action de l'humain. Les activités prédites sont calculées avec (1.20).

$$\begin{aligned} s_i(t) &= \sum_{p=1}^{n_{PC}} \omega_{pi}^S(t) \cdot PC_p(t) \\ S_i(t) &= V(t) \cdot S_i^d(t) + (1 - V(t)) \cdot \left(\frac{s_i(t)}{s\max(t)} \right) \end{aligned} \tag{1.20}$$

Dans une version simplifiée, on considère que lorsque le professeur interagit avec le robot, le signal de vigilance V devient égal à 1. Suivant la situation, avec interaction ou non ($V = 1$ ou $V = 0$), la sortie S_i sera soit le mouvement (orientation et vitesse) "désiré" $S_i^d(t)$ i.e. réalisé par le robot sous l'action de l'humain, soit le mouvement prédit par le modèle. Le poids ω_{pi}^S de la connexion entre la p^e cellule de lieu et la i^e action est adaptée selon le taux d'apprentissage $\epsilon(t)$ et la règle d'apprentissage (1.21) inspirée de la règle de descente de gradient de [Widrow and Hoff, 1960] :

$$\frac{d\omega_{pi}^S}{dt} = (S_i^d(t) - s_i(t)) \cdot PC_p(t) \cdot V(t) \cdot \epsilon(t) \tag{1.21}$$

La sortie directement prédite $s_i(t)$ est normalisée $s\max = \max_{i=1..n_S}(s_i)$ avec n_S la taille du champ moteur. Cette normalisation assure que la dynamique de la bulle gaussienne du DNF codant la direction de mouvement prédite est indépendante de la quantité d'apprentissage. En effet, le contrôle moteur qui est réalisé dépend du profil d'activation du DNF. En particulier, le DNF utilisé pour représenter l'orientation peut être dérivé pour obtenir la commande de rotation angulaire. La normalisation est donc importante pour que la dérivée soit encore forte même si le facteur d'apprentissage $\epsilon(t)$ est faible pour permettre un moyennage des orientations apprises.

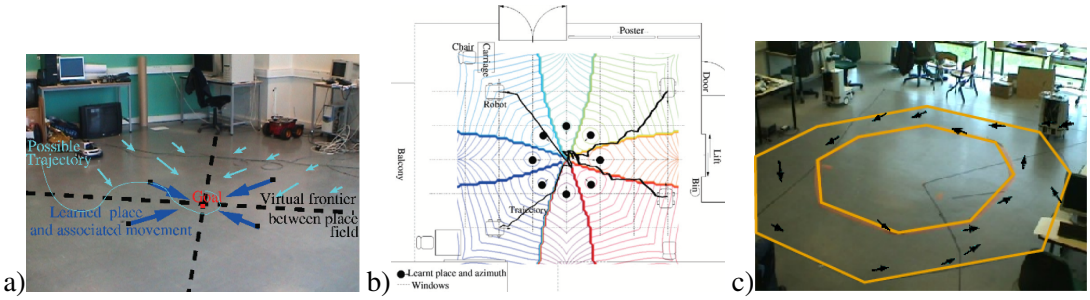


FIGURE 1.10 – Exemples de bassins d'attraction générés par des couples sensorimoteurs appris par un robot mobile. Chaque flèche représente une correction appliquée par le professeur en un lieu donné (cercle). Le robot apprend l'association entre ce lieu et l'orientation suivie. A chaque instant la reconnaissance des lieux prédit la direction à suivre. (a,b) Un bassin d'attraction basé sur quelques couples lieu/action suffit à générer le comportement pour rejoindre un lieu particulier de l'environnement. (c) Grâce à l'apprentissage en interaction avec un professeur humain, 3 tours d'apprentissage sont suffisants pour que le robot devienne autonome. Le professeur n'a plus besoin de corriger le comportement du robot. Le bassin d'attraction construit est stable et robuste aux perturbations (obstacles, "téléportations" i.e. déplacements sans retour proprioceptif...).

La combinaison de plusieurs couples associant un mouvement avec une région de l'environnement permet de contraindre un bassin d'attraction sensorimoteur. Ce bassin d'attraction peut définir un lieu à atteindre ou une trajectoire à suivre (Figure 1.10).

Dans le modèle complet, la détection de l'action de l'humain est remplacée par une détection de nouveauté [Giovannangeli and Gaussier, 2010]. Cette détection permet de séparer, d'une part, une correction du comportement visant à raffiner l'orientation suivie par le robot dans un lieu donné (adaptation de l'orientation) et, d'autre part, le besoin de recruter une nouvelle cellule de lieu quand aucun lieu n'est suffisamment bien reconnu. L'erreur de prédiction compare l'orientation prédite avec l'orientation suivie, intégrée sur le déplacement du robot entre deux cellules de lieu. La détection de nouveauté se fait par comparaison entre l'erreur de prédiction de l'orientation avec un seuil d'erreur qui peut être appris [Giovannangeli, 2007]. Ce principe de comparaison sur l'erreur de prédiction sera repris et modifié au chapitre 4 pour permettre d'adapter un comportement sans recruter de nouveaux couples lieu/action.

1.4.4 Comparaison entre PerAc et GMM/GMR sur une l'apprentissage du suivi d'un chemin

Nous avons aussi comparé le modèle PerAc et l'algorithme GMM/GMR [Calinon et al., 2009] dans le cas d'une expérience de navigation simple où le robot doit reproduire une trajectoire montrée par un professeur humain. Les résultats, publiés, sont présentés en annexe A. Le modèle PerAc permet d'apprendre et d'adapter rapidement le comportement du robot durant l'apprentissage de la trajectoire. De son côté, l'algorithme GMM/GMR permet d'apprendre une trajectoire plus souple que celle obtenue avec le modèle PerAc. Cependant, cet algorithme nécessite plusieurs démonstrations complètes des trajectoires à réaliser ce qui est assez contraignant pour le professeur humain. En pratique, nous avons réutilisé les trajectoires obtenues lors de l'apprentissage avec le modèle PerAc. Nous avons ensuite discuté la complémentarité des deux approches en terme de réactivité d'apprentissage versus la qualité de la trajectoire re-

produite. Ces deux contraintes semblent contradictoires. Il est probable que les deux approches correspondent à des stratégies présentes et agissant en parallèle dans le cerveau afin que l'individu puisse faire face aux différentes situations. PerAc pourrait représenter un apprentissage rapide en un coup alors que GMM/GMR correspondrait à une adaptation plus lente du modèle (différence d'échelles de temps). Quoiqu'il en soit, ces différents algorithmes ne traitent pas des problèmes réels que pose l'apprentissage de tâches complexes. Dans la section suivante, nous allons nous focaliser sur les problèmes spécifiques aux tâches et aux séquences "complexes".

1.5 Vers l'apprentissage de tâches complexes

Notre contrôleur sensorimoteur doit permettre à un robot d'interagir avec des humains pour apprendre des tâches complexes impliquant des séquences de gestes ou d'actions en fonction de buts et de motivations multiples. D'autre part, cette interaction devrait être naturelle c'est-à-dire correspondre à la manière d'interagir avec un animal ou un très jeune enfant. Nous nous situerons à un niveau pré-verbal de l'interaction, et nous nous intéresserons donc plus particulièrement aux comportements d'imitation sans utilisation explicite du langage.

1.5.1 Séquences fixées, séquences variables, buts et plans

Dans la section précédente, nous nous sommes intéressés à la construction de bassins d'attraction pour définir des comportements de suivi de trajectoires et d'atteinte de lieux. Des tâches plus complexes peuvent impliquer l'apprentissage de séquences fixes ou l'apprentissage de cartes cognitives permettant de réaliser des séquences variables dépendant des buts appris.

Apprentissage de séquences fixées

Différents travaux se sont focalisés sur l'apprentissage de séquences temporelles (cf. [Kühn and Hemmen, 1991] pour une revue). Dominey[Dominey, 1995] propose une architecture apprenant des séquences temporelles basée sur un modèle des structures corticales et sous corticales impliquant à la fois le cortex préfrontal et les ganglions de la base. Cette architecture permet d'apprendre des séquences de saccades visuelles. Le modèle s'appuie sur un réservoir de dynamique, fixé a priori, et un apprentissage de la sortie motrice. Les activités dans le réservoir de dynamique, correspondant à l'état du système présent au niveau préfrontal, évoluent dynamiquement grâce à des connections récurrentes. Suivant des approches similaires, le réservoir de dynamique peut être construit à partir d'autres solutions par exemple un réseau chaotique [Quoy et al., 2001; Daucé et al., 2002; Li et al., 2008] ou un Echo State Network (ESN, [Jaeger, 2001; Lagarde, 2010]).

L'un des inconvénients des séquences exclusivement temporelles est la difficulté de garder synchronisé le comportement et la situation réelle du robot. Ainsi, l'un des enjeux dans l'apprentissage de séquences est aussi la détection des conditions pour démarrer ou arrêter une séquence ou une sous séquence [Boucher and Dominey, 2006]. L'apprentissage des différentes relations entre les différents éléments de la séquence s'approche de la notion d'apprentissage d'une syntaxe comme celle qui pourrait être utilisée pour le langage. Les travaux [Boucher and Dominey, 2006] réalisés sur l'apprentissage des séquences ont ainsi été étendus à l'apprentissage du langage [Dominey and Boucher, 2005]. Le langage permet ensuite de faciliter l'apprentissage d'une

tâche par démonstration en permettant de guider le robot dans ces actions par des commandes vocales [Boucher and Dominey, 2006; Dominey et al., 2007].

Afin d'aller vers plus de collaboration entre l'humain et le robot, la notion de plan partagé est abordée dans [Lallee et al., 2009, 2010]. La séquence apprise est réalisée par deux individus. Ainsi, le robot apprend la séquence complète en même temps que la nécessité de partager les rôles. Lors de la reproduction du plan, le robot commence par déterminer quel rôle il va jouer dans la réalisation du plan. Dans [Bicho et al., 2010], les auteurs proposent une architecture permettant de réaliser un plan partagé pour assembler un objet en s'appuyant sur des champs de neurones dynamiques (DNF, Sec. 1.3.1). Les DNF apportent de la stabilité dans la prise de décision et permettent aussi de mémoriser les informations. Différents champs et leurs interactions permettent ainsi au robot d'inférer l'objet qui doit être assemblé, d'anticiper sur certaines actions ou de soulever un problème quand la séquence ne correspond pas à ce qui est attendu. Cependant, dans ces travaux, le plan n'est pas appris mais donné a priori.

Les solutions d'apprentissage de séquence que nous venons de décrire s'appuient sur des primitives motrices définies a priori. Ces primitives représentent des comportements minimaux que le robot doit connaître pour pouvoir apprendre des comportements plus sophistiqués. Cependant la question de l'apprentissage de ces primitives n'est clairement pas abordée. D'autre part, la notion de but n'apparaît que de manière implicite dans ces modèles : le but est la réalisation de la séquence apprise. De manière alternative, il est possible d'avoir toutes les séquences présentes dans un même graphe rassemblant donc les différentes actions possibles. Le comportement désiré s'appuie alors sur l'utilisation des buts appris pour sélectionner et planifier les actions adéquates. Il existe diverses solutions permettant de planifier des actions s'appuyant sur des planificateurs à plus ou moins haut niveau[Zacharias, 2012; Guitton, 2010]. Certaines solutions prennent aussi en compte des aspects d'interaction Homme-robot [Alami et al., 2005; Clodic et al., 2009].

Réalisation de séquences variables : planification avec des cartes cognitives apprises

Une solution basée sur des "cartes cognitives" apprises respectera mieux nos contraintes de plausibilité biologique et de construction des représentations. L'idée de carte cognitive a été émise par Tolman [Tolman, 1948] à partir d'expériences dans lesquels des rats traversent un labyrinthe. Une carte cognitive correspondrait à une représentation interne de l'environnement permettant à l'animal de trouver des chemins en prenant des raccourcis et en évitant les obstacles. Les cartes cognitives permettent aussi d'expliquer par un apprentissage latent la vitesse à laquelle le comportement du rat s'adapte lorsque une récompense (nourriture) commence à être distribuée. En effet, une telle modification du comportement peut difficilement être expliquée par des apprentissages stimulus-réponse classiques. Avec la découverte des cellules de lieu dans l'hippocampe [O'Keefe and Nadel, 1979], O'Keefe et Nadel ont pu proposer que les modèles de carte cognitive soient présentes dans cette structure cérébrale. On peut distinguer différents types de modèles de carte cognitive. Plusieurs modèles considèrent les cartes cognitives comme encodant les relations entres les différents lieux [Redish and Touretzky, 1998; Franz et al., 1998; Muller et al., 1996; Trullier et al., 1997]. Un parcours du graphe construit est alors nécessaire pour déterminer quel lieu rejoindre ensuite pour atteindre le but. Les processus pour réaliser ce parcours ainsi que pour générer les commandes motrices sont propres à chaque modèle mais ces processus sont généralement algorithmiques i.e. un système non neuronal est nécessaire pour extraire l'information pertinente de la carte cognitive. Dans [Voicu and Schmajuk, 2000], une

activité propagée de lieu en lieu depuis le but permet de déterminer quel est le lieu suivant à rejoindre. L'intérêt de cette méthode est de ne pas refaire le parcours complet a chaque changement de lieu du robot. En pratique, cette solution implique que le robot soit à chaque instant capable de connaître les lieux accessibles. [Arbib, 1981] suggère que cette connaissance des lieux n'est pas suffisante car, pour être utiles, les cartes cognitives doivent être centrées sur l'encodage des séquences d'actions permettant de passer d'un lieu à un autre. Ainsi, les cartes cognitives doivent correspondre à l'apprentissage de la topologie de l'espace i.e. des connections entre les différents lieux existants. En proposant un modèle de l'hippocampe prédisant les évènements multimodaux, Banquet et collègues [Banquet et al., 1997; Revel et al., 1998] introduisent l'idée des cellules de transition et d'une carte cognitive utilisant ces transitions. Les cellules de transition sont un support biologique pour coder l'information des changements de lieu possibles i.e. pour estimer les lieux accessibles depuis un endroit donné. On introduit alors une organisation dans la structure des cartes cognitives : les cellules de lieu donnent naissance aux cellules de transition qui permettent ensuite de construire les cartes cognitives. Des transitions pourront aussi être associées avec des buts créés permettant la planification du comportement. Nous reviendrons ci-après sur ce modèle qui sera utilisé et complété par la suite dans cette thèse. Enfin, [Hasselmo, 2005; Martinet et al., 2011] proposent aussi d'apprendre les cartes cognitives au niveau cortical. Dans ces modèles, les colonnes corticales représentent des états (lieux), des actions (transitions) ou des buts et sont intercalées suivant le schéma : état-action-état-action-état-but.

Le modèle de cartes cognitives que nous utiliserons dans cette thèse est le modèle développé au sein du laboratoire. Il s'agit d'un modèle inspiré de l'Hippocampe et du cortex préfrontal [Banquet et al., 1997]. Ce modèle a été implémenté en simulation avant d'être utilisé sur un robot réel [Gaussier et al., 2001; Banquet et al., 2005; Cuperlier et al., 2007; Hirel et al., 2011]. L'architecture utilisée (Figure 1.11) permet d'apprendre des transitions entre lieux, de construire des cartes cognitives reliant ces transitions et de planifier les déplacements grâce à cette carte. Les activités des cellules de lieu sont calculées suivant l'information visuelle comme expliqué auparavant. Une compétition (*Winner-Takes-All*, WTA) sur les cellules de lieu permet de déterminer la cellule de lieu la plus reconnue assimilée au lieu où se trouve le robot. Lors des déplacements, le robot change de lieu ce qui sera détectable par un changement de cellule de lieu gagnante. L'information du lieu gagnant et du changement de lieu est présente au niveau du cortex entorhinal (EC) en entrée de l'hippocampe. Dans le modèle de l'hippocampe, le rôle

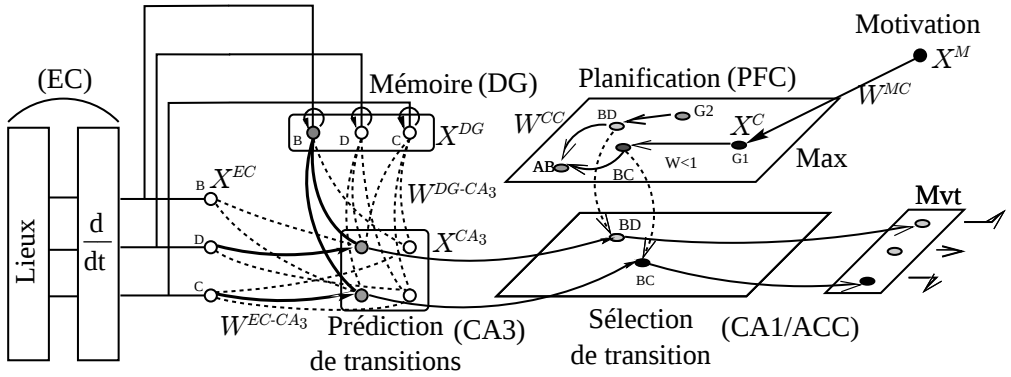
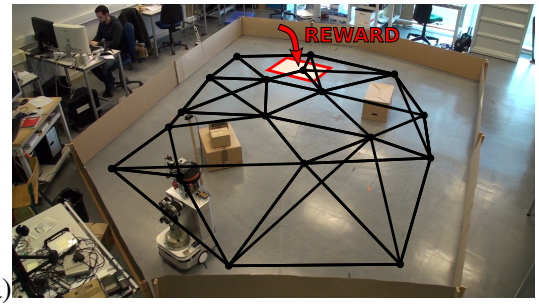


FIGURE 1.11 – Schéma du modèle de boucle hippocampo-corticale pour la planification.

simplifié du Gyrus Dentelé (DG) est de mémoriser l'information du lieu passé. Les cellules pyramidales de CA3 de la Corne d'Ammon apprennent alors l'association entre le lieu passé et le lieu présent. Elles fournissent ainsi une information prédisant les transitions possibles depuis un lieu donné. Ce modèle simple de l'hippocampe est suffisant pour apprendre des séquences spatiales et temporelles [Lagarde et al., 2008; Hirel et al., 2010, 2013]. Il peut néanmoins être complété par la construction d'un état interne permettant de discriminer les sous séquences qui se répètent [Lagarde et al., 2007]. Les transitions créées peuvent aussi être utilisées pour construire une carte cognitive.

Le principe de la carte cognitive est assez simple, les transitions entre les lieux appris dans l'environnement sont liées entre elles par des connexions synaptiques au niveau du cortex préfrontal et forment une carte topologique de l'environnement. Lorsqu'un but est découvert, la connexion entre le neurone qui code le contexte motivationnel pour rejoindre ce but et le neurone représentant la dernière transition est renforcé. Ainsi l'activité liée au contexte motivationnel se propage à travers la carte cognitive en diminuant à chaque connexion synaptique (dont les poids sont inférieurs à 1). L'activité d'un neurone est calculée comme la valeur maximale des activités post-synaptiques qu'il reçoit. Ainsi, selon la position et l'orientation de la transition par rapport au but, son activité dans la carte cognitive sera plus ou moins grande. Cette activité permet de sélectionner la transition à réaliser donnant l'orientation à suivre. On peut alors trouver le chemin - topologique - le plus court vers le but par une simple remontée de gradient.

a) 

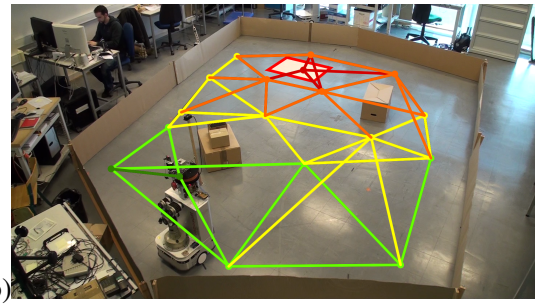
b) 

FIGURE 1.12 – Représentation dans l'espace 2D de la carte cognitive construite en ligne durant la tâche de navigation visuelle avec une plate-forme mobile Robulab (Robosoft). (a) Le graphe des transitions est construit. Une récompense est reçue à un endroit de l'environnement. Le besoin satisfait par la récompense est associé à la dernière transition réalisée. (b) Lorsque le même besoin est actif, il excite dans la carte cognitive un potentiel d'activité qui est propagé de transition en transition suivant le graphe construit (gradient du vert vers le rouge). Dans un lieu donné, le robot compare les potentiels de chaque transition possible pour choisir le chemin à suivre. Il rejoindra ainsi le lieu récompensé en suivant le plus court chemin topologique.

La figure 1.12 présente un exemple de carte cognitive apprise et utilisée. Le robot explore son environnement et apprend de manière latente un modèle topologique, i.e. une carte cognitive, de son environnement suivant les changements de lieux se produisant. La carte cognitive est donc construite et encodée sous la forme d'un graphe dont les noeuds sont les transitions entres les cellules de lieu. Suivant l'approche animat [Wilson, 1991; Meyer, 1996], le robot peut aussi découvrir des ressources. Ces ressources permettent de satisfaire un besoin physiologique comme de la nourriture satisfaisant la faim. Leur découverte est alors traité comme une récompense par le robot. Lors de la satisfaction d'un besoin, la dernière transition réalisée est associée dans la carte cognitive avec le drive correspondant. Un drive est activé à la fois quand le besoin

est satisfait et quand il se fait sentir. Cette transition devient alors un but qui sera activé par le drive associé. Le robot se dirigera, suivant le plus court chemin, vers l'endroit où il avait pu satisfaire son besoin.

Les cartes cognitives permettent de gérer différents contextes motivationnels (besoins) et différents buts possibles et de planifier les déplacements du robot. Le fait que l'apprentissage de la carte cognitive soit latent assure une meilleure réactivité comportementale du robot dès la première réception d'une récompense dans l'environnement. Les capacités de planification incluent la possibilité de replanifier par exemple si un obstacle empêche de prendre le chemin choisi. Nous verrons au chapitre 3 que le principe des cartes cognitives peut être utilisé en dehors du contexte de la navigation. Je présenterai dans ce chapitre le détail du modèle de carte cognitive tel qu'il a été utilisé dans les expériences. Nous nous intéresserons en particulier à la gestion des buts utilisés par la carte cognitive afin de faciliter l'apprentissage de tâches par démonstration ou en interaction sociale avec un humain.

Dans cette section, nous avons dressé un panorama rapide des techniques basées sur des réseaux de neurones permettant d'apprendre et de reproduire des tâches pouvant impliquer des séquences fixes (timing appris) mais aussi des séquences variables (liées au plus court chemin) dépendant de buts appris. Nous allons maintenant nous intéresser à la manière d'apprendre ces tâches en situation d'interaction.

1.5.2 Apprentissage par imitation

Les espèces sociales, telles que les grands singes, les humains ou les oiseaux, peuvent combiner de multiples stratégies quand elles sont engagées dans des processus d'apprentissage de nouveaux comportements. C'est le cas aussi bien quand le sujet tente de résoudre une tâche par lui même à partir de précédentes observations, que quand il est impliqué dans une interaction plus coopérative avec un parent ou un démonstrateur [Tomasello et al., 2005]. Par exemple, il a été montré que les singes sont capables de tirer avantage de différentes stratégies que ce soit : l'exploration, le "stimulus enhancement" (augmentation de la saillance d'un stimulus) [Spence, 1937], la facilitation de réponses [Whiten and Ham, 1992] via l'amorce des activités neuronales et la reconnaissance des affordances [5], ainsi que de l'émulation [Tomasello et al., 1987] ou encore des formes simples d'imitation [Byrne and Russon, 1998]. Chez l'Homme, et plus particulièrement durant les 4 premières années du développement, nous savons que les mécanismes mentionnés précédemment sont exploités en conjonction avec les fonctions de communication [Uzgiris, 1999] : la mise en place d'un tour de rôle [Nadel et al., 2005] permet une alternance d'essais et de corrections (le professeur imitant souvent son élève). Mais il n'est pas clair si ces mécanismes suffisent à expliquer les comportements d'imitation tels que l'imitation spontanée (imitation de gestes dans un but de communication [Andry et al., 2001]), l'imitation différée (aussi appelée apprentissage par observation [Thorndike, 1898]) i.e. la capacité à reproduire un ensemble donné d'actions pour remplir un objectif à partir de l'observation d'un professeur [Piaget, 1945b]), l'apprentissage par démonstration, ou encore la coopération [Zukowgoldring and Arbib, 2007]. Si l'imitation est connue comme un marqueur central de l'épigénèse, elle reste en même temps un terme générique et flou [Jacob and Jeannerod, 2005]. L'imitation revêt en effet de multiples formes comme illustré dans la figure 1.13 au travers de trois situations d'apprentissage qui seront considérées dans cette thèse. L'imitation peut se faire en miroir, c'est-à-dire

5. qualité des systèmes suggérant leur propre utilisation

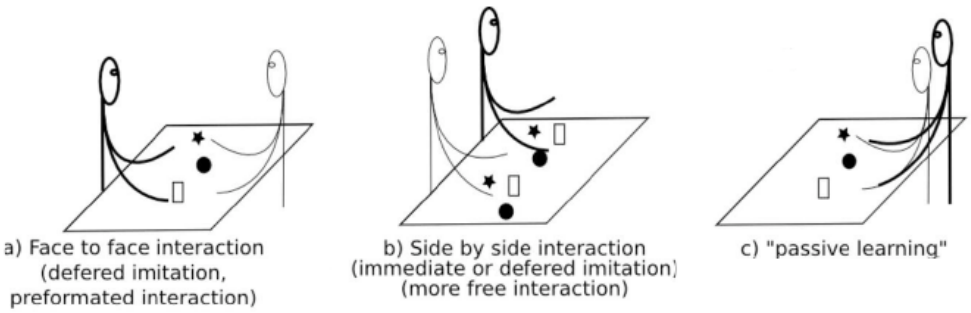


FIGURE 1.13 – Quelques situations d'apprentissage par imitation.

que les agents sont face à face, côte à côte, ou bien le professeur peut venir guider physiquement les mouvements de l'apprenant (démonstration passive). Dans ces trois situations apparaissent deux composantes : une composante spatiale dans la relation entre l'apprenant et le professeur (i.e. leur position relative), et une composante temporelle. Selon cette composante temporelle, l'interaction peut se faire immédiatement et dans ce cas, les mêmes mouvements sont reproduits en même temps par les deux agents, ou bien la reproduction peut être faite en différé. Dans ce cas, l'apprenant observe ce que fait le professeur et réalisera les mêmes gestes ou actions a posteriori. Nous distinguerons aussi l'imitation de la posture et l'imitation des conséquences sensorielles produites par les gestes imités. En effet, l'imitation de la posture soulève le problème de la correspondance [Nehaniv and Dautenhahn, 2002] i.e. la difficulté de faire correspondre les mouvements de l'imitateur (robot) aux mouvements de l'imité (humain), compte tenu des morphologies parfois différentes et de la perception souvent limitée de la posture de l'autre. Résoudre le problème de la correspondance n'apparaît pas comme nécessaire pour obtenir des capacités d'imitation (des conséquences sensorielles). Nous considèrerons donc que la résolution de ce problème sera réalisée à une étape postérieure non traitée du développement de notre robot.

Neurones miroirs et imitation

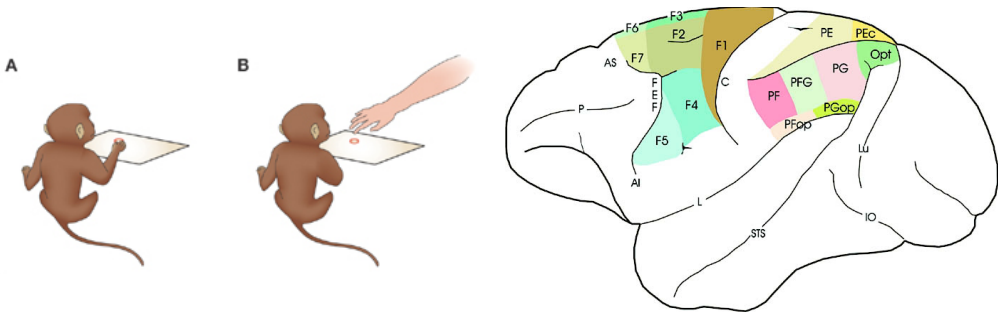


FIGURE 1.14 – Les neurones miroirs sont des neurones qui ont la particularité de s'activer à la fois quand l'action est réalisée (A) et quand elle est observée (B). Ces neurones ont été découverts [Gallese et al., 1996] chez le singe dans l'aire F5 du cerveau qui correspond au cortex prémoteur. Image issue de [Rizzolatti et al., 2009]

Les neurones miroirs sont considérés comme une explication possible des capacités d'imitation. Rizzolatti et al. [Gallese et al., 1996; Rizzolatti and Craighero, 2004] ont découvert chez les singes des neurones qui s'activent à la fois quand le singe réalise une action et quand il observe un autre singe ou un humain en train de réaliser la même action. Ces neurones appelés *neurones miroirs* ont originellement été trouvés dans l'aire F5 du cortex prémoteur. Dans les singes, les activations des neurones miroirs sont liées à des actions telles que attraper et placer des objets. Une action observée peut produire une activation même si elle est partiellement cachée. Ainsi dans [Gallese et al., 1996], les auteurs émettent l'hypothèse que le rôle des neurones miroirs est de permettre la reconnaissance des actions réalisées par les autres. Suivant le phénomène de résonance, la représentation motrice de l'observateur sera activée par les gestes observées. L'observateur peut alors se référer à son "savoir-faire moteur" pour reconnaître et comprendre les actions qu'il a vues. L'approche exclusivement centrée sur les neurones miroirs a progressivement évolué vers le principe d'un "système miroir" réparti sur plusieurs structures corticales [Rizzolatti and Craighero, 2004]. Les preuves en faveur d'un système miroir réparti sur les aires corticales sont nombreuses. Le Lobule Pariétal Inférieur (IPL) présente aussi des neurones dont l'activation sur le schéma des neurones miroirs tandis que dans le Sulcus Temporal Supérieur (STS), certains neurones sont sensibles à l'observation d'actions réalisées par les autres et donc semblent être un support pour permettre l'apprentissage par imitation [Oztop et al., 2006]. Cependant, le système miroir est-il vraiment limité aux quelques structures corticales mentionnées. Dans une perspective intégrée de l'ensemble des structures cérébrales, nous devons aussi considérer le rôle des structures sous-corticales dans le développement des comportements sensorimoteurs, et leur rôle possible dans le système miroir.
Suivant la théorie du système miroir, le développement d'un tel système serait une adaptation liée à l'évolution qui permettrait d'améliorer les capacités d'imitation et donc la collaboration sociale. Plusieurs modèles ont donc utilisé cette hypothèse pour justifier des capacités de reconnaissance d'actions et d'apprentissage par imitation et les implémenter dans des robots [Demiris, 2002]. Cependant, le rôle des neurones miroirs dans les tâches d'apprentissage peut être remis en question car il n'est pas évident de trancher si ces neurones sont une cause ou une conséquences du développement des compétences. Même si une proto-imitation de la protrusion de la langue a été observé chez les nouveaux nés [Meltzoff and Moore, 1977; Kugiumutzakis, 1998], cela ne permet pas de conclure que les capacités d'imitation sont innées [Jones, 2009].

Imitation cause ou conséquence du développement ?

Depuis plusieurs années, différents auteurs [Gaussier et al., 1998; Metta et al., 2006; Heyes, 2010] ont défendu que les neurones miroirs sont un effet de bord des apprentissages sensorimoteurs. Plusieurs travaux récent ont proposé des modèles visant à expliquer comment les neurones miroirs peuvent émerger à partir de l'apprentissage d'associations sensorimotrices [Metta et al., 2006; Nishimoto et al., 2008]. Ces travaux se focalisent sur l'émergence des capacités de reconnaissance des actions observées. Dans [Gaussier et al., 1998], les auteurs ont montré que des comportements d'imitation peuvent être obtenus à l'issue d'un simple apprentissage d'associations sensorimotrices. Sans être capable de reconnaître les actions des autres, ni même de reconnaître l'autre de soi-même, un robot peut déjà exhiber un comportement d'imitation de bas niveau [Andry et al., 2004].

1.6 Conclusion

Dans ce chapitre, nous avons vu que bien qu'ayant de multiples avantages pour décrire le contrôle moteur chez l'humain et pour contrôler des robots (sec. 1.1), le contrôle optimal reste limité pour rendre compte des aspects d'interaction sociale (sec. 1.1.4). La définition du comportement en terme de fonction de coût et d'optimisation est attrayante pour simplifier la représentation et l'inclusion de contraintes sur le mouvement. Cependant, cette définition ne permet pas d'envisager les adaptations et les modulations à très bas-niveau du contrôle moteur comme observé chez l'humain dans les expériences d'interaction sociale.

Puis, notre brève étude (sec. 1.2) des structures cérébrales et des apprentissages dans le cerveau a confirmé que si l'optimisation peut faire partie des mécanismes implémentés par certaines structures cérébrales, le cerveau ne peut se résumer à un tel contrôle moteur. L'approche dynamique des comportements (sec. 1.3) insistent sur l'importance de définir des attracteurs sensorimoteurs pour construire des comportements robustes et stables mais surtout profiter "gratuitement" d'un certain nombre de propriétés émergentes en termes de mémoire dynamique ou de sélection de l'action, comme c'est le cas avec les DNF (sec. 1.3.1). Cependant, les DNF présentent différents inconvénients liés à la difficulté à adapter en ligne le contrôle réalisé. Les paramètres définissant le noyau gaussien doivent être soigneusement choisis en fonction de la tâche à réaliser. Un compromis est nécessaire entre la zone d'action des attracteurs générés et la vitesse pour rejoindre ces attracteurs. Les travaux récents sur l'apprentissage dans des DNF [Detorakis and Rougier, 2012] permettront peut être de lever ces limitations.

Dans la suite, nous utiliserons le modèle PerAc (sec. 1.4) pour construire des bassins d'attractions grâce à des apprentissages sensorimoteurs. Les comparaisons avec LWPR [Vijayakumar et al., 2005] et GMM/GMR [Calinon et al., 2009] (Sections 1.4.2 et 1.4.4) ont montré que les résultats étaient comparables avec un avantage pour le modèle PerAc dans le cas d'un apprentissage rapide et en ligne. Enfin, le contrôle PerAc s'approche d'un contrôle respectant le principe du minimum d'intervention [Todorov and Jordan, 2003]. En effet, le robot peut s'écarter de la trajectoire désirée, la direction du robot ne sera corrigée que s'il s'éloigne trop i.e. s'il risque de sortir du bassin d'attraction appris. Ainsi, un contrôleur de type Optimal Feedback Control [Todorov and Jordan, 2002] n'est pas nécessaire pour mettre en place le principe de minimum d'intervention. Suivant l'approche PerAc, la dynamique sensorimotrice du contrôleur, apprise en interaction avec un humain, est suffisante pour obtenir ce principe. De plus, les cartes cognitives peuvent compléter le modèle PerAc pour inclure l'apprentissage de but et la planification dans les comportements appris (sec. 1.5.1).

Enfin, nous avons vu dans la section 1.5.2 que l'imitation prend de multiples formes rendant difficile sa définition. Cependant, certaines capacités d'imitation peuvent résulter d'apprentissages sensorimoteurs simples. De plus, si nous voulons construire des robots capables de s'adapter à des situations d'interaction dyadiques (Homme et robot) ou triadique (Homme, robot et objet), nous devrons mettre en place une architecture aussi polyvalente. Suivant une approche épigénétique, une telle architecture impliquera de prendre en compte le contrôle moteur [Ijspeert et al., 2003; Schaal et al., 2004], la fusion multimodale et la coordination sensorimotrice (particulièrement la vision et la proprioception) [Andry et al., 2004; Chaminade et al., 2008; Montesano et al., 2008], l'apprentissage de séquences temporelles [Lagarde et al., 2007; Calinon et al., 2007; Demiris and Johnson, 2003], les buts, les récompenses et l'information contextuelle. Ceci soulève la question de l'existence d'un nombre limité de mécanismes

ou principes qui permettraient d'obtenir gratuitement à partir d'un même modèle l'ensemble des capacités d'apprentissage et de communication issues des comportements d'imitation. Nous proposerons qu'un tel modèle puisse être construit à partir de plusieurs mécanismes associatifs simples, permettant de prendre en compte les multiples modalités (commandes motrices, proprioception des articulations, stimuli visuels, besoins internes, récompense, état émotionnels, etc.) du système cerveau-corps-environnement.

Dans le chapitre suivant, je présenterai mon modèle de contrôle qui s'attaque à la construction d'attracteurs sensorimoteurs pour le contrôle d'un bras robotique en s'appuyant sur les principes de l'approche PerAc.

La connaissance s'acquiert par l'expérience, tout le reste n'est que de l'information.

– Albert Einstein

CHAPITRE 2

Apprentissage visuomoteur du contrôle d'un bras robotique et attracteurs comportementaux

Nous avons vu précédemment qu'une solution classique pour le contrôle moteur d'un bras robotique en situation d'interaction exploitait un contrôle en impédance (Sec. 1.1.2). Ce contrôle s'appuie sur une dynamique du second ordre générée en comparant position et vitesse actuelles du bras avec une position et une vitesse désirées. Les points de cette trajectoire peuvent être extraits par des méthodes statistiques à partir de données d'entraînement (GMM, DMP, LWPR). L'approche Perception-Action (Sec. 1.4) permet d'apprendre en ligne des comportements en s'appuyant sur des catégorisations sensorielles associées à des actions motrices. Ces associations génèrent dynamiquement des bassins d'attraction définissant des trajectoires ou des points fixes à rejoindre. Cette approche permet un apprentissage en ligne, en situation d'interaction, des comportements. Dans ce chapitre, nous chercherons à combiner la gestion de la dynamique motrice d'un bras robotique telle que réalisée par un contrôle en impédance avec la souplesse d'apprentissage qu'apporte l'architecture PerAc.

Dans un premier temps (cf. article 1, Sec. 2.1), je présenterai les différentes étapes aboutissant à la mise en place du modèle Dynamic Muscle PerAc (DM-PerAc) qui s'appuiera sur un modèle simple des muscles pour réaliser un contrôle en impédance et qui pourra en même temps apprendre des bassins d'attraction dynamiques. Suivant l'approche PerAc, le contrôleur de bras de robot pourra découvrir les associations entre les entrées sensorielles et les activations motrices permettant de générer les bassins d'attraction utilisés pour contrôler les mouvements du robot. Nous étudierons dans quel espace encoder les commandes motrices et donc les attracteurs. En effet, une transposition naïve du modèle PerAc à l'espace moteur du bras à N degrés de liberté consisterait à travailler sur des catégories fusionnant les N degrés de liberté du bras. Cette approche ne pouvant pas donner de résultats satisfaisants pour maintenir une posture suite à un apprentissage rapide, le contrôle sera réalisé articulation par articulation en s'appuyant sur des activations musculaires correspondant aux raideurs du système {ressorts,amortisseur} simulant les muscles (Fig. 2.1). Chaque articulation sera donc contrôlée suivant une équation dynamique du second ordre simulant l'influence des contractions musculaires d'un muscle agoniste et d'un muscle antagoniste. Ce modèle de contrôle s'approchera d'un contrôle en impédance, à la différence qu'il permettra de générer des bassins d'attraction dynamiques sur des postures mais

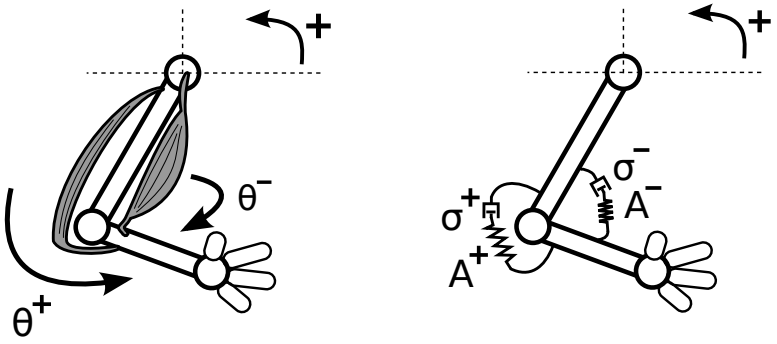


FIGURE 2.1 – Modèle simplifié du contrôle d'une articulation basée sur des muscles agonistes et antagonistes approximés par un ressort et un terme d'amortissement. Les propriétés d'amortissement sont supposées être des propriétés mécaniques des muscles bien que liés à la raideur des muscles.

aussi sur des trajectoires. Nous nous intéresserons ensuite à l'apprentissage d'attracteurs posturaux dans l'espace moteur. Les mouvements du bras induiront des changements perceptifs qui seront utilisés pour modifier la contraction des muscles afin de corriger les mouvements produits. Ce principe permettra au modèle DM-PerAc d'apprendre des homéostats proprioceptifs en adaptant les coefficients du contrôle moteur (activations musculaires). Afin d'obtenir un contrôleur visuo-moteur, ces homéostats seront ensuite associés à des catégories visuomotrices fusionnant catégories visuelles et catégories proprioceptives. L'activation des catégories visuomotrices induira l'activation d'une combinaison d'attracteurs qui déterminera le mouvement. Ce contrôleur visuomoteur permettra des comportements d'imitation de bas niveau et sera utilisé au chapitre 3 comme base pour le développement d'apprentissages interactifs de tâches plus sophistiquées.
Dans le même article, je présenterai comment exploiter au sein du modèle DM-PerAc le principe "Yuragi" , proposé par Fukuyori et collègues [Fukuyori et al., 2008]. En particulier, le principe du "Yuragi", basé sur l'équation de Langevin, permettra d'améliorer la précision du contrôle sans nécessiter de réapprentissage.

Dans un second temps (Sec. 1.3.1), nous étudierons les propriétés dynamiques de notre modèle en le comparant avec les champs de neurones dynamiques (DNF, Dynamical Neural Fields). Les DNF présentent plusieurs propriétés intéressantes notamment une capacité de bifurcation entre deux attracteurs possibles et un effet de mémoire permettant d'avoir une hystérésis dans la prise de décision. Nous proposerons une méthode pour évaluer le comportement généré par notre contrôleur et le représenter suivant une base similaire à celle des Champs de Neurones Dynamiques (i.e. profil d'activité de la population de neurones). Pour finir, nous utiliserons cette représentation commune pour montrer que notre contrôleur possède la propriété de bifurcation.

2.1 Article1 : Construction d'un contrôleur visuomoteur pour bras de robot suivant l'approche PerAc

de Rengervé, A., Andry, P., and Gaussier, P. (2013b). On-line Learning and Control of different kinds of attraction basins for development of sensory-motor control strategies. *Biological Cybernetics*, page soumis

Biological Cybernetics manuscript No.
(will be inserted by the editor)

On-line Learning and Control of Attraction Basins for the Development of Sensorimotor Control Strategies

Antoine de Rengervé · Pierre Andry · Philippe Gaussier

Received: date / Accepted: date

Abstract Imitation and learning from human require an adequate sensorimotor controller to learn and to encode the behaviors. We present the Dynamic Muscle PerAc (DM-PerAc) model to control a multi DOF robot arm. In the original Perception-Action (PerAc) model, path following or place reaching behaviors correspond to the sensorimotor attractors resulting from the dynamics of learned sensorimotor associations. The DM-PerAc model, inspired by human muscles, permits to combine impedance-like control with the capability of learning the sensorimotor attraction basins. We detail a solution to incrementally learn on-line the DM-PerAc visuomotor controller. Postural attractors are learned by adapting the muscle activations in the model depending on movement errors. Visuoarticular categories merging visual and proprioceptive signals are associated with these muscle activations. Thus, the visual and proprioceptive signals activate the motor action generating an attractor which satisfies both visual and proprioceptive constraints. This visuomotor controller can serve as a basis for imitative behaviors. Besides, the muscle activation patterns can define directions of movement instead of postural attractors. Such patterns can be used in state-action couples to build attractor trajectories like in the PerAc model. We discuss a possible extension of the DM-PerAc controller by adapting the Fukuyori's controller based on the Langevin's equation. This controller can serve not only to reach attractors which were not explicitly learned but also to learn the attractor trajectories.

Keywords visuomotor control · impedance control · perception-action loop · neural network

ETIS UMR CNRS 8051, ENSEA, University Cergy Pontoise F-95000 Cergy Pontoise, France
E-mail: {rengerve,andry,gaussier}@ensea.fr),

1 Introduction

In order to act efficiently in unknown environment and collaborate with human, robots must be able to control and to adapt their behaviors. On the contrary of classical motor control approach, Human-Robot Interaction and imitation paradigms take into account that a human partner can influence and improve both the behavior and the behavioral learning of a robot. Our past work, following a developmental approach [40], along with collaborations with developmental psychologists, cognitive psychologists and neurobiologists have led us to understand that the tasks and behaviors cannot be reduced to a set of controlled parameters. Behaviors rather emerge from the dynamics of perception action coupling [21, 41]. The behavior is built upon a wide range of interactions at different levels. A behavior learning system must be able to capture the dynamical *sensorimotor attractors* describing the behaviors. Their parameters are - unfortunately partly- linked to the control architectures, but also to the agent's embodiment, and to the interactions of this brain-body system with the surrounding environment. In such conditions, the issues of learning, adapting and sharing these attractors are fundamental in order to achieve natural and intuitive non verbal human-robot interaction. What are the constraints on the low level motor control to learn such attractors? What kind of model of motor control should be used and how can it be learned ?

Impedance control enhances optimal control in the case of interaction with the environment (Sec. 2.1). In impedance control, position and velocity constraints determine the movements with respect to the desired trajectory. In the framework of human robot interaction, regression based solutions [31, 8] can learn the desired trajectories from data obtained during the task demonstration by a human (Sec. 2.2). The trajectories result from mixtures of adapted kernels. Impedance control can be linked to muscle activations (Sec. 2.3). Though, the hypothesis of a desired trajectory is usually kept while focusing on the link between muscle activations and the impedance control parameters (stiffness,...). On the contrary, we defend the Perception-Action (PerAc) approach claiming that behaviors correspond to sensorimotor attractors emerging from the dynamics of multiple learned sensorimotor associations (Sec. 3.1).
In our first works on the emergence of imitation [22, 3], we showed that an arm controller using the learning of visuoarticular associations to build an homeostatic controller can lead to the emergence of low level imitative behaviors if the perception is ambiguous (i.e. when mistaking partner's hand for its own hand). However, this visuomotor controller had several limitations. In particular, it did not allow the coding of attractor trajectories by state-action couples like in the PerAc approach. We thus propose, in this paper, a model called Dynamic-Muscle PerAc to control a robot

arm with multiple Degrees-of-Freedom (Sec. 4). The DM-PerAc model is based on simple models of muscles and joints with dynamic equations corresponding to impedance control. This DM-PerAc model learns the inverse kinematic model by learning visuoarticular associations. It also learns postural attractors to link perception (visuoarticular categories) with actions coded as muscle activations i.e. it also learns the inverse dynamic model. The behavior and properties of the DM-PerAc visuomotor controller are evaluated in Section 5. Like in our previous works [3], the DM-PerAc visuomotor controller is a good bootstrap for imitative behaviors (Sec. 6.2). Besides, the muscle activation patterns can be used in state/action couples to code trajectories like in the PerAc model (Sec. 6.1). We introduce the Fukuyori's controller to improve performance in Section 6.3 and we discuss its possible role to learn trajectories with the DM-PerAc model in Section 7.

2 State of the art of on-line, incremental motor control for learning from interaction

2.1 Impedance control

In optimal control theory [54], the desired trajectory is an optimal trajectory crossing given via-points and minimizing some movement variables like jerk[1] [16]. The motor control should be flexible enough to allow physical interaction with the environment. Studies of movement properties have led to impedance control model [29] as an approximation of neuro-muscular properties. According to the equilibrium trajectory hypothesis [15], motor programs are internally represented as the trajectories of an equilibrium point. Impedance control is efficient to control manipulators acting in contact with the world [13]. Impedance control is also a usual controller for prostheses and exoskeleton which involve direct physical interaction with a human [35]. Impedance control is based on a second order "damped mass spring"-like system (1) enabling constrained motion, dynamic interaction and obstacle avoidance.

$$M\frac{dV}{dt} = K(X_0 - X) + B(V_0 - V) \quad (1)$$

with V the velocity and X is the Cartesian position of the end effector. The coefficient K (equivalent to the spring stiffness) and B represent the constraints related to the position command X_0 and the speed command V_0 respectively. Some other versions of impedance control use the proprioceptive information (e.g. [1]) instead of the Cartesian position. Besides, the via-points, which are necessary to compute the desired trajectory $(X_0(t), V_0(t))$, can be learned from watching [42].

[1] In the minimum-jerk approach, the movements maximize the smoothness of the motion.

2.2 Learning tasks from human with regression techniques

The trajectories can be directly learned from training data obtained during a task demonstration by a human. In order to learn how to fulfill a task, a human teacher can provide feedback or data which are integrated in a sensorimotor model of the task. Function approximation based on local regression techniques [5] is efficient to learn forward or inverse models of robot control. Learning an initial model from a human demonstration reduces the size of the space to be explored. Demonstrations facilitate and improve a subsequent reinforcement learning [49]. More recent, the Locally Weighted Projection Regression algorithm (LWPR) [55] merges both the incremental learning properties of the Receptive Field Weighted Regression (RFWR) algorithm [51] and the projection of input data in order to reduce the dimensionality problem. The authors showed a demonstration with a 30-DOF SARCOS humanoid robot learning the dynamic inverse model and performing eight-shaped trajectories with its arm.

Regression techniques to learn models of motor control were also used in learning from demonstration paradigm [4]. The Dynamic Movement Primitives (DMP) [31, 50, 28, 32] are based on the RFWR algorithm. The primitives are control policies that are activated depending on a local basis function. They provide motor control as a second order dynamic system. The combination of primitives shapes the attractor landscape to produce the desired trajectory. This combination depends on a phase variable which gives the temporal reference of the movement. The approximated function is the time-dependent trajectory, and locally weighted regression of training data determines the parameters of the basis functions (number, centers, bandwidths) and the contribution of corresponding primitives. The DMP algorithm shows interesting properties of spatial and temporal invariance and was applied to learn discrete and rhythmic movements. However, the correspondence problem [43] was completely eluded as the training data were obtained from joint-angle recording system on the human. A particular coupling must be introduced in the dynamic equation of the phase variable in order to tackle correctly perturbations. The action of this coupling is to slow the evolution of the phase variable when there are perturbations.

Similarly, a Gaussian Mixture Model (GMM) can also learn a model of a demonstrated task by encoding proprioceptive and Cartesian information in Gaussian kernels [8]. The learning is based on an Expectation-Maximization process which adapts the Gaussian kernels to describe probabilistically the input data obtained in a training session. Then, given partial information like only the Cartesian position, Gaussian Mixture Regression extracts the probable proprioception to control a robotic arm. Depending on the task, vision or motion capture devices can track particular ele-

ments (e.g. spoon, human head) [10, 11]. Still, the computation of the 3D Cartesian coordinates of the visual markers requires particular calibrations of the external devices. [9] uses a dynamical second order motor controller and Hidden Markhov Models (HMM) instead of GMM. HMM encodes the sequential dependencies in the task, whereas the motor controller now implements impedance control. A trade-off between the position constraint and the speed constraint is managed depending on the variance in the demonstrated trajectories. This version of the model is similar to DMP. The main difference is that the learning of the constraints on the position and the velocity profile can take into account the mutual influence between different Degrees-of-Freedom which is not the case with DMP. Some recent works [38, 47] studied the on line adaptation of the control stiffness from the position variations and haptic feedback. This adaptation of the control improved the quality of the collaboration between Human and robot [47].

In the aforementioned solutions, the control depends on a desired trajectory statistically estimated from the demonstrations done by a human. The learning process requires training data from several demonstrations and is often performed off-line in a batch mode. This is a limitation to build a reactive and interactive system. During the trajectory learning, the system is passive. In other approaches, like in the Perception-Action approach [21], the robot must act in order to learn. As the actions of the robot can be directly monitored and adapted, behaviors can be learned on-line in situation of interaction. The motor control must be able to generate actions without knowing the desired trajectory which is only available after the learning.

2.3 Adaptation of muscle activations and impedance control

In the case of human arm control, the actions are generated by muscle contraction. The VITE model [6] is based on equations describing the muscle activations. The resulting dynamics is similar to the dynamics produced by an impedance controller [26]. However, the VITE model also assumes a target position to drive muscle activations. In iterative and adaptive control [53], the behavior can be adapted by changing the control parameters instead of changing the command. Considering the adaptation properties at the level of muscular control [7, 17], the authors proposed a muscle centered model of adaptive and iterative control to maintain a posture or to follow a trajectory under disturbances [20]. The controller takes into account a feedforward torque command and a feedback control to generate the final torque command. The feedforward torque command is generated by muscular activation. The feedback controller is proportional derivative. Such control can be equivalent to impedance control if the apparent inertia is assumed to vary and to be

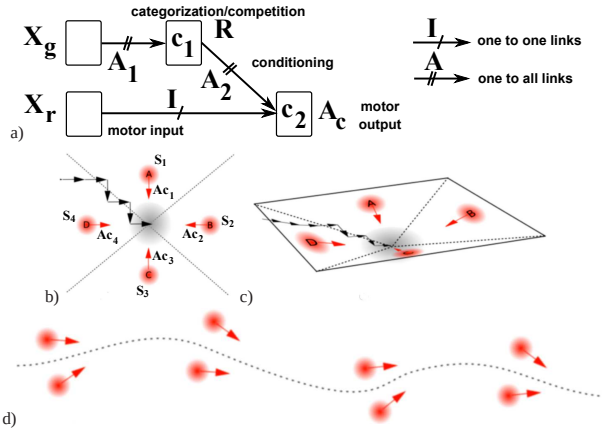


Fig. 1 *a)* PerAc model. *b-d)* Example of dynamical attractors in 2D space. *b,c)* Fixed point attractor. *d)* Trajectory.

equal to the inherent inertia of the robot. The muscle activations are adapted in order to reduce the feedback error. Indeed, in the model [20], the adaptation of the muscle activities directly induces changes of the feedforward torque and of the stiffness in the feedback controller. Feedforward torque modification enables to compensate an applied external force. In the case of rapidly varying disturbances, the stiffness of the feedback controller is increased, so the robustness of the controller also increases. However, increasing the stiffness from a muscular point of view is energy consuming. So, the stiffness will tend to decrease when the unpredictable perturbations cease to occur. This model permitted to maintain a desired posture or to follow an a-priori given trajectory. The principle of adapting the muscle activations should not be reduced to adapting the parameters of the impedance control. This principle is also interesting to learn the perception-action coupling.

3 The Perception-Action model and arm control

3.1 Sensorimotor associations to learn on-line dynamic attraction basins and low-level imitative behaviors

Since many years, we have defended the Perception-Action approach (PerAc, [21]) claiming that, in an active system, coupling perception and action enables to build behaviors. Fast on-line learning of associations between sensory signals and motor signals is sufficient to build sensorimotor attraction basins. Let us consider the sensorimotor system of an agent acting in a given environment (or state space) and having 2 sensation vectors $\mathbf{X}_r$ and X_g (Fig. 1a). First, $\mathbf{X}_r$ represents the proprioception, the coarse feedback information from the execution of the motor command or the direction of the goal (if the goal is in immediate neighborhood). It can be considered as the reflex or the regulatory pathway

that links a proprioceptive sensor to the motor command **Ac**. Second, $\mathbf{X}_g$ represents more global information about the environment and allows to build a local but robust distance measure (metric). This measure is learned and computed by a competitive recognition group **R**. The operator *c* represents a competitive structure (soft-WTA) able to self organize depending on one sensory data flow (categorization or conditioning). This basic sensorimotor architecture uses recognized place-cells (**R**) associated with orientation of movement (**Ac**). It provides an efficient control for navigation and path following [25]. Figure 1b-d shows examples of attraction basins defined in a 2D space. In Figure 1b and c, the Voronoi diagram shows for any point of the space which state is recognized, and also which action is performed. By doing these actions, the system behaves like it moves in an attraction basin to an attractor place (similarly, to an attractor trajectory in Fig. 1d). The attraction basins emerge from the system dynamics generated by state/action couples. The associations are learned on-line from interaction with a teacher [24]. When the robot moves away from the desired trajectory, the human teacher changes its orientation to correct its behavior. This feedback is used to learn new place-cell/orientation couples to complete the sensorimotor control and to modify the robot behavior. This sensorimotor learning enables the robot to follow trajectories and even to reach particular locations which become attractors for the dynamical system. In the PerAc approach, the perception is considered as the result of learning sensation/action associations allowing a globally consistent behavior while facing an object. For instance, by learning sensorimotor associations, a robot can learn how to return to a given object which can be interpreted as the fact that the robot "perceives" the object [41].

The same sensorimotor association principle can be a basis for the emergence of low level imitative behaviors [22]. In the case of arm control, we showed [3] that an imitation of directly observed gestures can appear as a side effect of a homeostatic visuomotor controller with perceptual ambiguity. During a first phase, the system learns associations between visual and motor signals building a visuomotor *homeostat*. Due to low visual capabilities, the robot is unable to discriminate its own hand from the hand of a teacher (ambiguity of perception). As the control architecture implements a homeostat, the system tends to maintain the equilibrium between visual and proprioceptive information. If a difference is perceived, then the system acts to come back to the equilibrium state. To do so, the robot moves its arm so that its proprioceptive configuration corresponds to the perceived visual stimuli according to its sensorimotor learning. As a result of these movements, the demonstrator's gestures are imitated [3]. The correspondence problem [43] is avoided as the robot only imitates what is observed with its own capabilities.

In the model of [3, 39], the control was performed in the visual space. A forward kinematic model allowed to estimate the visual position of the robot hand. This position was then compared with the perceived visual position to generate movements (see [3] for details). A first drawback was that erratic estimations of the visual position of the robot hand produced an erratic control. Because the forward model learning was based on Self-Organizing Maps [37], false estimations could occur until learning convergence. So, the controller should not be used before the end of learning. The learning process was not incremental. Finally, the trajectories were not coded by sensorimotor couples like in the PerAc model. Indeed, the motor commands were extracted from the Dynamic Neural Fields [52] by using an ad hoc readout mechanism. This solution presented interesting properties (memory, bifurcation) (see Sec. 5.3), but was only able to define attractor positions. Moreover, we were not able to explain how the readout process could be learned or tuned. Here, we are interested in a model that can bootstrap imitative behaviors and can also code trajectories according to the PerAc approach. The model should also be incremental and able to managed multiple Degrees of Freedom. The main issue is which coding should be used to represent motor actions. Taking inspiration from control in human, the muscles may play an important role.

3.2 Limitations of PerAc for the control of multiple Degrees-of-Freedom

We then studied how the PerAc model can serve to control a robotic arm. The difficulty is that arm control involves more Degrees-of-Freedom than navigation. In [34] [3], the authors developed arm controllers which work in spaces different from the motor space, reducing the number of dimensions. The difficulty is then to extract a motor command from the control in the lower dimension space. We use the alternate solution consisting in performing the control in the proprioceptive space. The generation of the motor command is simplified whereas the difficulty is to learn sensorimotor attractors. We compared two versions of the PerAc model in the case of postural attractor learning. The first model encodes both the sensory category and the command vector in the N-dimensional space (N being the number of DOF). The second model is equivalent to N PerAc controller building 1D sensorimotor attractors. We show that the correct model should be able to associate action vectors defining N 1D attractors with the direct sensory categorization (2) of the N-dimensional motor space.

The PerAc model has been adapted naively to learn attractors in N-dimensional space. The learned states categorize the proprioceptive configuration and are associated with movement vectors defined in the N-dimensional angular space. The proprioception is composed of the angular

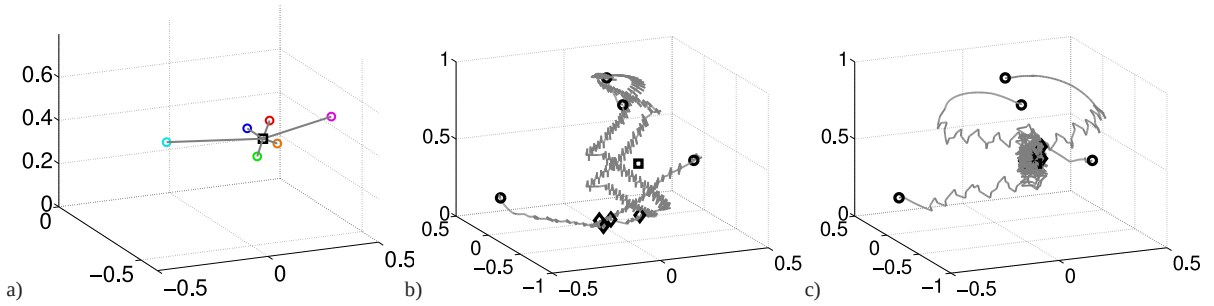


Fig. 2 Learning an attractor at a target posture (square) with a Katana robotic arm (4 DOF). *a)* Representation of the arm end effector position in the 3D Cartesian space. The attraction basin is learned from 6 demonstrations of corrective movements. The associated actions are learned as the difference vector between the initial posture and the target posture. *b)* Motor control with states and actions encoded in the 4D proprioceptive space. Four trajectories of the arm end effector in the Cartesian 3D space are plotted. Each trajectory starts from a different position (circle). Diamonds are the final positions. The arm end effector does not converge to the desired posture. In each case, the movements of the arm are stopped before the arm bump into its support. *c)* Same test with independent attractors built in each joint space. The arm end effector converges to a point near the desired posture.

positions of the controlled joints ($\mathbf{P} = [\theta_1 \ \dots \ \theta_N]$, index j)[2]. The categorization of the proprioceptive input is described by (2) and (3). The proprioceptive inputs $\mathbf{P}$ are encoded into categories $\mathbf{S}^P$ with Gaussian responses depending on a variance parameter β^P. The variance parameter β^P enables to increase or to reduce the selectivity of the sensory categories. They are recruited with a process based on Adaptive Resonance Theory [12]. If the current input $\mathbf{P}$ is too different from any encoded sensory pattern $\mathbf{W}_i^P$, i.e. if the recognition $\mathbf{S}_i^P$ is under a vigilance threshold λ^P, then a new category i_r is recruited ($\varepsilon^P = 1$). The current sensory input $\mathbf{P}$ is stored on the weights $\mathbf{W}_{i_r}^P$ to the i_r^{th} category. Even though a slow adaptation of the encoded categories is also possible, we do not consider it in this article.

$$\begin{cases} S_i^P = exp(-\frac{\sum_j (P_j - W_{ij}^P)^2}{2\beta^P}) \\ \Delta W_{i_r j}^P = \varepsilon^P \cdot (P_j - W_{i_r j}^P) \\ \text{with } \varepsilon^P = \mathscr{H}(\lambda^P - \underset{i}{max}(S_i^P)) \end{cases} \tag{2}$$

with the Heaviside function $\mathscr{H}(x) = 1$ if $x > 0$ and 0 otherwise. The recognition activities $\mathbf{S}^P$ are normalized to give the output of the recognition process $\mathbf{R}^P$ (3).

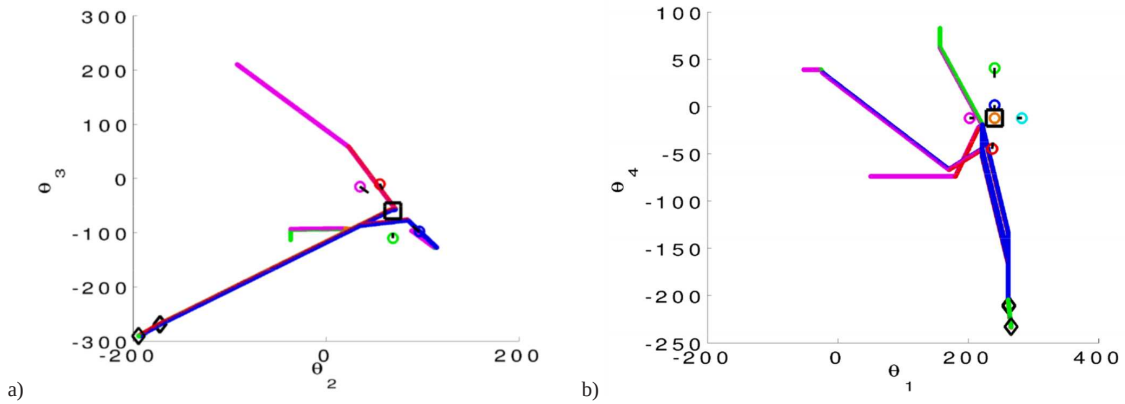
$$R_i^P = \frac{S_i^P}{\sum S^P} \tag{3}$$

The output R_i^P can be interpreted as the probability that the sensory category i is the current sensory state of the robot. In practice, we approximated the sensory categorization process to a winner-takes-all which corresponds to the variance parameter β^P tending to 0 i.e. the selectivity for the categories R_i^P is maximal. Each learned category i is associated with a movement vector $\mathbf{Ac}_i$. These vectors are difference vectors in the N-dimensional motor space between the target position and the encoded starting positions. Given the output activities $\mathbf{R}^P$, the motor command $\mathbf{Ac}$ (4) is equal to the difference vector $\mathbf{Ac}_{i_m}$ associated to the winning category i_m. The target position is not explicitly encoded in a category and should emerge as an attractor point due to the state/action couples pointing toward it.

$$\mathbf{Ac} = \sum_i R_i^P \cdot \mathbf{Ac_i} \tag{4}$$

Even if the encoding is done in the proprioceptive space, the interactions with human caregivers may rely on the 3D Cartesian space or on the visual space. We designed a simple experiment testing the reaching of a position learned from the demonstration of the movements correcting particular deviations from the target. To simplify the task, the position is demonstrated as a unique joint configuration. Six movements to correct the deviations are demonstrated. They start from six different initial points around the target (above, below, on the left, on the right, behind and in front). In each case, the demonstrated gestures bring the arm in the exact posture that should become an attractor (Fig. 2a). In order to focus on learning and testing the demonstrated attraction basin, we reduce the categorization phase to learning six proprioceptive categories encoding the six starting points of the demonstrated movements. Four trials with different starting points were performed to test the model. The experiments were realized with an electrical 4 DOF robotics arm (Katana from Neuronics AG.). The obtained trajectories are shown in Fig. 2b. None of the trajectories converged to the target position. In the experiment with robot, the arm was frozen before it bumped into its support. The learning protocol was reproduced in simulation. The motor state diverges showing that the encoded state/action couples do not define a postural attractor (Fig. 3). The encoded vectors (for the states and the actions) take all joints equally. A subset of the joints can bias the recognition of one specific state. However, as there is only one global movement vector associated with this state, performing the corresponding action can im-

[2] Bold letters indicates vectors whereas plain letters are scalars.



a) b)

Fig. 3 Simulation of the posture learning test in the case of the N-dimensional sensory categories and movement vectors. The learning protocol of Fig. 2 is reproduced in the simulation. The used colors match the colors of the states in Fig. 2a. *a)* Projection of the categories and the trajectories in the 2D space of the 2^{nd} and 3^{th} joints. *b)* Projection of the categories and the trajectories in the 2D space of the 1^{st} and 4^{th} joints. The alternation between several state/action couples with uncompensated movements makes the system diverge.

prove the situation for some of the joints but not for all. The same situation can happen for a few states that will alternatively direct the arm movement. As the different consecutive commands are not perfectly compensated, the trajectories diverge. Increasing the number of learned state/action couples may improve the constraints on the attraction basin. Yet, the learning of these supplementary state-action pairs would be expensive and tedious to demonstrate for a human.

We now present the alternate strategy which consists in splitting the problem into learning N attractors in independent 1D spaces (one for each joint). The different state/action couples are thus encoded in each joint space. The states $S^J_{j,i}$ encode positions $W^J_{j,i}$ of the j^{th} joint. At each moment and for each joint j, the state $R^J_{j,i}$ is only active when the i^{th} state is maximally recognized for the j^{th} joint (5).

$$\begin{cases} \forall j,\ S^J_{j,i} = (W^J_{j,i} - \theta_i)^2 \\ R^J_{j,i_{max}} = 1 \ \text{ if } i_{max} = \underset{i}{argmin}(S^J_{j,i}) \\ \qquad\qquad 0 \ \text{ otherwise} \end{cases} \tag{5}$$

For each joint, the states are associated with 1-Dimensional movement vectors Ac_j. The resulting motor command **Ac** is the combination of the different movement vectors Ac_j of the joints (6).

$$\begin{cases} Ac_j = \sum_i R^J_{j,i} \cdot Ac_{j,i} \\ \mathbf{Ac} = [Ac_1, \ldots, Ac_N] \end{cases} \tag{6}$$

The experiment protocol of Fig. 2a is kept, resulting in the trajectories of Fig. 2c. The encoded state/action couples define an attractor near the target. The oscillations around the attractor point are due to the discretization of the speed command and the lack of anticipation in the controller. However, this second strategy fails to learn and reproduce particular trajectories resulting from 1D state/action couples. The multiple couples can combine into pairs inducing parasite postural attractors instead of only shaping the movements to reproduce the trajectory. To prevent this ill-suited pairing, the learned 1D state/action couples should be associated with categories in the N-dimensional space representing the state of the system. Each time a desired movement is shown (e.g. by manipulating the arm), N 1D state/action couples are learned to store the desired movements for each of the N joints. These couples should be associated with the N-dimensional category representing the configuration when the demonstration occurred. Then, the learned movement would only be reproduced when the state of the system is similar to the learned configuration. In this approach, coding a particular movement for the N joints requests only N 1D state/action couples (one for each joint) whereas maintaining a particular posture requests at least 2N couples because 2 couples are needed for each joint. The motor controller should be able to learn either a particular movement or a postural attractor. In the next section, we describe the Dynamic-Muscle PerAc (DM-PerAc) model which provides a common coding basis for both aspects of the control. The DM-PerAc model is based on a simplified model of joints and muscles where both particular movements and postural attractors are coded as muscular activations.

4 Dynamic-Muscle PerAc model

We now present our model called Dynamic Muscle PerAc to control a robotic arm. This model combines control equivalent to impedance control with the PerAc principle. The parameters and equations of the DM-PerAc model are all summarized in Appendix A.

4.1 Control of joint position with a simplified muscle model

Different models like Hill's model [27] and Huxley's model [30] have been developed describing different properties of the muscles. In the lumped-parameter nonlinear antagonistic muscle model [56, 57], the movements of a joint are produced by a couple of antagonist muscles. The muscles are simulated by Hill's muscle model. This model is based on three components: a contractile element, a series elastic element and a parallel elastic element. In [36], the two elastic elements are neglected to focus on the dominant contractile element. The contractile element can be approximated by a force generator in parallel with a damping element [19]. The force generator implements the force-length relation in muscles with the force that can be modulated by neural signals [57]. The damping element implements the force-velocity relation given by [27].

Our model, called Dynamic-Muscle PerAc (DM-PerAc), is also based on couples of antagonist muscles (hereafter noted $+$ and $-$) around the joints with each muscle approximated as a contractile element. However, unlike [36] and [57], we use a simplified linear model of contractile element which generates torque instead of force. In the DM-PerAc model, the torque generator is a spring with variable stiffness whereas the damping element is a simple viscous damper (Fig. 4). The varying stiffness is given by the muscle activations **A**. The joint positions are controlled with the equations [7-13]. As these equations are the same for each joint, the joint index j is not displayed. Besides, the time step (t) dependency is only indicated to disambiguate terms when different time steps are involved in the same equation. For each joint, the agonist and the antagonist muscles generate the apparent torques τ^+ and τ^- (7).

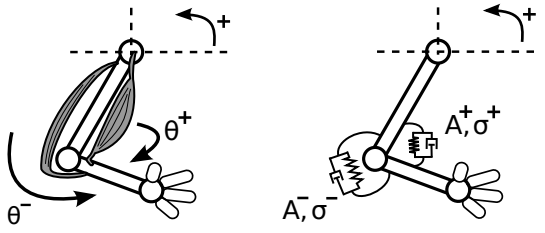


Fig. 4 Simplified model of muscle control relying on a spring damped model of muscles. Damping properties are hypothesized as mechanical property of the arm still related to the muscle stiffness.

$$\begin{cases} \tau^+ = -A^+ \cdot \theta^+ - \sigma^+ \cdot \dot{\theta}^+ \\ \tau^- = -A^- \cdot \theta^- - \sigma^- \cdot \dot{\theta}^- \end{cases} \quad (7)$$

where A^+ (resp. A^-) is the muscle activation and σ^+ (resp. σ^-) is the damping factor[3] of the agonist (resp. antagonist) muscle. The angular values θ^+ and θ^- are measured respectively from the full flexion position θ_{max} and from the full extension position θ_{min} (8).

$$\theta^+ = \theta - \theta_{max} \ , \quad \theta^- = \theta - \theta_{min} \ \text{ and } \ \theta \in [\theta_{min}, \theta_{max}] \quad (8)$$

with θ the angular position of the joint.

The dynamical equation of the system links the rotational acceleration $\ddot{\theta}$ and the moment of inertia I with the torques generated by the agonist and antagonist muscles given by (7).

$$I \cdot \ddot{\theta} = \tau^+ + \tau^- = -A^+ \cdot \theta^+ - \sigma^+ \cdot \dot{\theta^+} - A^- \cdot \theta^- - \sigma^- \cdot \dot{\theta^-} \quad (9)$$

Equations (8) and (9) gives the equation (10) where $\sigma = \sigma^+ + \sigma^-$:

$$I \cdot \ddot{\theta} = A^+ \cdot (\theta_{max} - \theta) - A^- \cdot (\theta - \theta_{min}) - \sigma \cdot \dot{\theta} \quad (10)$$

The system defines an attractor at the convergence point $\theta_{eq} = \frac{A^+ \cdot \theta_{max} + A^- \cdot \theta_{min}}{A^+ + A^-}$. To simplify this controller, the angular positions θ of the joint are normalized so that for each joint, they vary between 0 and 1.

$$\theta_{min} = 0 < \theta < \theta_{max} = 1 \ , \ \theta^+ = 1 - \theta \ \text{ and } \ \theta^- = \theta \quad (11)$$

Equation (10) then simplifies into (12) with $\theta_{eq} = \frac{A^+}{A^+ + A^-}$ with $K = A^+ + A^-$.

$$\ddot{\theta} = \frac{K}{I} \cdot (\theta_{eq} - \theta) - \frac{\sigma}{I} \cdot \dot{\theta} \quad (12)$$

The equation (12) corresponds to a classical mass-spring-damping system with a stiffness K and an equilibrium position θ_{eq}. The equilibrium position is unchanged when both A^+ and A^- are multiplied by the same factor. Such a factor only modifies the equivalent stiffness K. An adaptation of the stiffness K and the damping factor σ controls the rise time, overshoot and settling time. The controller was simulated using discrete time with a time increment Δt. With I the moment of inertia and τ the sum of the torques $\tau = \tau^+ + \tau^-$, the equations of the dynamical system are:

$$\begin{cases} \theta_t = \theta_{t-\Delta t} + \dot{\theta}_t \cdot \Delta t \\ \dot{\theta}_t = \dot{\theta}_{t-\Delta t} + \ddot{\theta}_t \cdot \Delta t \\ \ddot{\theta}_t = \tau_t / I \end{cases} \quad (13)$$

The variables $\theta_t, \dot{\theta}_t, \ddot{\theta}_t$ correspond respectively to $\theta, \dot{\theta}, \ddot{\theta}$ in the equations [7-12].

[3] The damping factor can be constant. However, controlled movements are improved if the damping factor varies with the stiffness. For instance, the damping term can be defined as proportional to the square root of the stiffness like in [20].

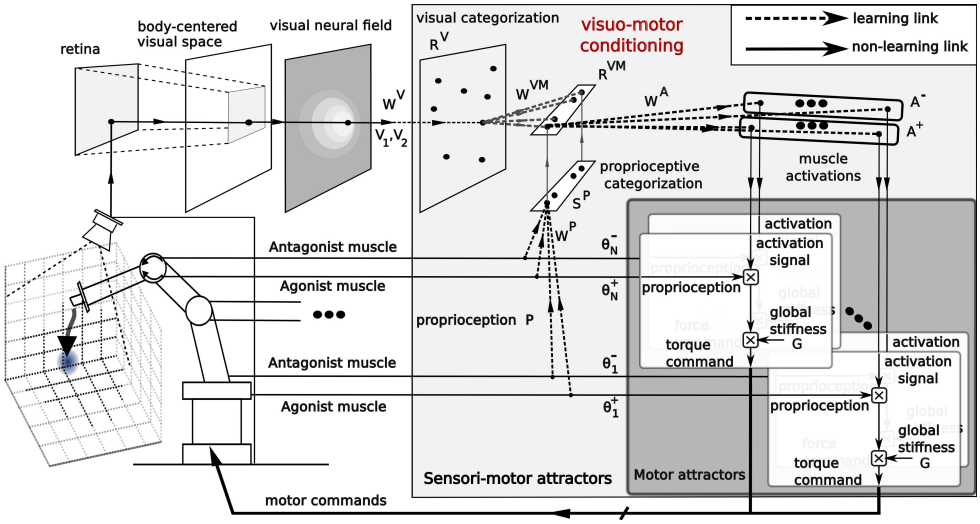


Fig. 5 Architecture of the visuomotor arm controller. Both visual and proprioceptive information are categorized. The visual input is associated with the proprioceptive input. The visuoarticular categories are then associated with the muscle activations defining the motor attractors. The visual input activates the associated visuoarticular categories and thus the corresponding motor attractors.

4.2 DM-PerAc visuomotor controller

The DM-PerAc model can use the previously described simplified muscle model with learned visuoarticular associations to build a visuomotor controller (Fig. 5). Visual and proprioceptive signals are merged into visuoarticular categories which are associated to the muscular activations determining the arm movements i.e. defining postural attractors. We will present first how the visuoarticular categories are built from learned visual and proprioceptive categories. Then, we will detail how the postural attractors are learned as muscle activations associated with the visuoarticular categories.

Proprioceptive categories are recruited during a motor babbling process in accordance with the categorization equations (2). Considering the agonist/antagonist paths, the proprioceptive information is defined by $\mathbf{P} = [\theta_1^+ \dots \theta_N^+ \; \theta_1^- \dots \theta_N^-]$ (index m). Each value $\theta^{+/-}$ is positive and normalized and follows the agonist or antagonist references (see Fig. 4). In our robotic setup, the visual information is captured by a single camera. A visual feature detector (e.g. color detector) enables to extract points of interest. The information is then projected over two 1D fields or vectors using population coding. Each vector codes the accumulated salience for the projected points of interest. The retina-centered vectors are then converted into body-centered vectors by a transformation using the pan and tilt angles of the camera. The body-centered vectors are computed as Dynamic Neural Fields [52]. Thus, they exhibit bifurcation and memory properties which are interesting in this attentional processing context. The coordinates (v_1, v_2) of the maximally salient point in this field are considered as the visual input. The visual categories are updated and learned using the equations (14) based on the equations (2).

$$\begin{cases} R_k^V = \frac{S_k^V}{\sum S^V} \text{ with } S_k^V = exp(-\frac{\sum_l (V_l - W_{kl}^V)^2}{2\beta^V}) \\ \Delta W_{kl}^V = \varepsilon^V \cdot (V_l - W_{kl}^V) \\ \text{with } \varepsilon^V = \mathscr{H}(\lambda^V - \max_k(S_k^V)) \end{cases} \tag{14}$$

The recruitment of a visual category R_k^V increases the vigilance threshold λ^P of the proprioceptive categorization in order to facilitate the recruitment of a proprioceptive category if none already encodes the current posture.

The visual and proprioceptive signals are merged in a visuoarticular layer. There is a bijection between the proprioceptive categories and the visuoarticular categories. Whenever a new proprioceptive category is recruited, a new visuoarticular category S_i^{VM} is also recruited and associated with it. The visuoarticular category is then associated with the muscle activations $\mathbf{A}$ maintaining the categorized posture. The aim of the visuoarticular learning process is to determine which visual category R_k^V is maximally activated when the arm reaches the attractor posture S_i^P. The connection weights W_{ik}^{VM} are increased depending on the co-activated visual (R_k^V) and proprioceptive (S_i^P) categories (15):

$$\Delta W_{ik}^{VM} = \varepsilon^{VM} \cdot S_i^P \cdot (f(S_i^P) \cdot f(R_k^V) - W_{ik}^{VM}) \tag{15}$$

with ε^{VM} a constant learning rate. The function f is defined by $f(X_l) = 1$ if $X_l = \max_l(X_l)$ and $f(X_l) = 0$ otherwise. The co-activation is only learned when the arm is close enough to the posture S_i^P, so the learning is modulated by the factor S_i^P that checks if the similarity measure S_i^P is high enough. Incorrect visuoarticular associations can be progressively forgotten.

The activities of the neurons in the visuoarticular layer are computed with the following equations (16):

$$\begin{cases} R_i^{VM} = \frac{S_i^{VM}}{\sum S^{VM}} \quad \text{with} \quad S_i^{VM} = R_i^P \cdot \sum_k (g(W_{ik}^{VM}) \cdot R_k^V) \\ g(W_{ik}^{VM}) = \quad 1 \text{ if } \left(\frac{W_{ik}^{VM}}{\max_k(W_{ik}^{VM})} \right)^n > 0.5 \\ \qquad\qquad\quad 0 \text{ otherwise} \end{cases} \tag{16}$$

A weight W_{ik}^{VM} contributes either as a factor 1 or 0 in the update equation. The connection with maximal weight, among the input connections to a neuron i, always gives a factor equal to 1. Other connections can be "active" (factor equal to 1) if their weights are close enough to the maximum. Several visual categories can then activate the same visuoarticular category. The normalization of the activities of the visual categories R_k^V ensures that the activities of the visuoarticular categories S^{VM} are always smaller than 1. The saturation of the neural activities is thus avoided. Besides, when the exponent n tends to $+\infty$ only the connection with maximal weight is equal to 1 and any others are null. We consider this particular case in the experiments.

The learning is performed on-line and fast. It is also incremental. By modifying some parameters (vigilance λ^P/λ^V or variance β^P/β^V) of the sensory categorization process, new visual and proprioceptive categories can be added on-line and are directly available for the visuomotor control. The vigilance parameter determines how much categories can overlap. Increasing the vigilance, i.e. allowing more overlapping, will increase the number of recruited categories. The variance parameter of the Gaussian kernels can be decreased with a similar result. If the variance is reduced, the selectivity of the categories increases and more categories will be recruited. Maintaining the vigilance level enables to maintain a certain level of overlapping and thus of interferences during learning.

As a result of a visuomotor association learning, a visual input can elicit visuoarticular categories which activate motor actions (muscle activations) to drive the arm to the proprioceptive configuration associated with the visual constraint. When a new visuoarticular category is recruited, the muscle activations enabling to maintain the visuoarticular configuration (in practice maintaining the proprioceptive configuration is enough) are learned. Muscle activation coefficients are learned on-line in a perception-action process. The sensory-motor loop is essential. As the system acts, it corrects or modifies its motor commands on-line to maintain the desired posture of the arm. The corrective movements are learned by reinforcing the adequate connection weights to the muscle activation neurons $\mathbf{A} = [A_1 \ldots A_{2N}] = [\mathbf{A}^+, \mathbf{A}^-]$. The activities of the visuoarticular categories $\mathbf{R}^{VM}$ determine the muscle activations $\mathbf{A}$ with (17):

$$A_m = \sum_i W_{mi}^A \cdot R_i^{VM} \tag{17}$$

where the weight W_{mi}^A is the learned activation of m^{th} muscle to maintain the arm in the proprioceptive configuration i. In the neural implementation of DM-PerAc model, the muscle activations $\mathbf{A}$ are bounded ($\mathbf{A} \in [0,1]^N$) due to the weight learning (see below (19)). Hence, the muscle activations $\mathbf{A}$ are multiplied by a constant stiffness factor G increasing the amplitude of the apparent stiffness. For each joint j, a noise term η_j is also added in the motor command to produce various movements helping the learning of the muscle activations. The previous dynamic equation (10) becomes (18):

$$I_j \cdot \ddot{\theta}_j = G \cdot (A_j^+ \cdot (\theta_{j,max} - \theta_j) - A_j^- \cdot (\theta_j - \theta_{j,min})) - \sigma_j \cdot \dot{\theta}_j + \eta_j \tag{18}$$

The resulting equilibrium point is unchanged whereas the apparent stiffness is now equal to $G \cdot K$.

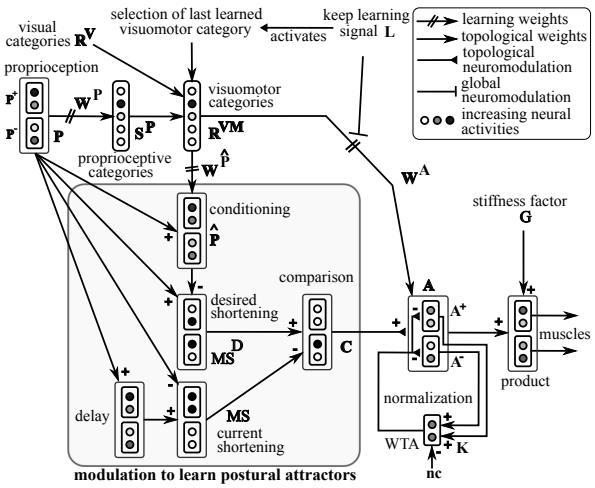


Fig. 6 Neural network learning the muscle activations to maintain the robotic arm in desired proprioceptive configurations. Learning is based on a neuromodulation process reinforcing the weights W_{mi}^A so the muscle activations $\mathbf{A}$ enable to maintain the desired posture. A second neuromodulation loop induces the normalization of the stiffness $\mathbf{K}$ of the different joints to avoid saturating the muscle activations.

The learning of the weights W_{mi}^A is performed just after the recruitment of the visuoarticular categories. The learning equation (19) is based on two terms and one learning factor:

$$\begin{aligned} \Delta W_{mi}^A = \mathscr{H}(L - th_L) \cdot (& \varepsilon^A \cdot C_m \cdot R_i^{VM} \cdot (1 - W_{mi}^A) \\ & - \alpha^A \cdot W_{mi}^A \cdot \max_j [K_j - nc]^+) \end{aligned} \tag{19}$$

where ε^A is a learning rate, α^A is a decay rate and $[x]^+ = x$ if $x > 0$ and 0 otherwise. As a result of the learning factor ($\mathscr{H}(L - th_L)$), the muscle activations are only modified during a limited period of time depending on the signal L. This signal L evaluates the need to continue adapting the muscle activations (see below, (23)). During this time, only the last recruited visuoarticular category i_r is active ($R_{i_r}^{VM} = 1$

and $R^{VM}_{i\neq i_r} = 0$). So, the learned muscle activations are only associated with this category. The positive term in (19) increases the muscle activations thus changing the attractor so that it matches the desired posture. This adaptation is based on the correction signal C detailed below (20). The second term in (19) is negative. Its role is to normalize the stiffness K_j of the joints to the constant value nc[4]. This normalization is necessary to avoid the saturation of both the weights W^A_{mi} and the neural activities A_m which would prevent any further correction of the movements.

The part of the architecture in the gray rectangle in Figure 6 is dedicated to the computation of the correction signal C. For each joint, the signal C compares the desired movements $\mathbf{M}^D$ with the current movements $\mathbf{M}$ (20) to determine if a muscle should contract more i.e. increase the corresponding muscle activation.

$$C_m = \mathscr{H}(M^D_m - M_m) \tag{20}$$

Each neuron in the desired movement layer M^D evaluates the need to contract the muscle m ($M^D_m = 1$ or 0) to correct posture. To do so, the equation of M^D_m (21) determines if the muscle "length" P_m (i.e. θ^+ or θ^-) should be reduced to match the desired "length" $\hat{P}_m$.

$$M^D_m = \mathscr{H}(P_m - \hat{P}_m - th_D) \tag{21}$$

where th_D is a threshold under which no correction is necessary. The desired position $\hat{P}_m$ is learned in one shot by associating $\mathbf{P}$ to $R^{VM}_{i_r}$ when the i^{th}_r visuoarticular category is recruited. The corresponding update and learning equations are (22):

$$\hat{P}_m = \sum_i W^{\hat{P}}_{mi} \cdot R^{VM}_i \quad \text{with} \quad W^{\hat{P}}_{mi_r} = P_m - W^{\hat{P}}_{mi_r} \tag{22}$$

The correction signal C_m (20) does not change the muscle activations if the current movement M_m already reduces the muscle length error by decreasing P_m. This information is computed by $M_m(t) = \mathscr{H}(P_m(t-\Delta t) - P_m(t))$ with $M_m = 1$ when no change of the muscle activation should occur.

The muscle activations for a given visuoarticular category are learned during a variable period of time depending on the comparison between the "learning enabling" signal L and the threshold th_L. The learning continues as long as there is unexpected correction of the muscle activations (23). Unexpected correction is determined by comparing for each muscle occurring correction C_m with its prediction $\hat{C}_m$. The occurrence of an unexpected correction increases the value of the signal L, thus extending the learning time period.

$$L(t) = \mathscr{H}(L(t-\Delta t) - th_L) \cdot \sum_m [C_m - \hat{C}_m]^+ + \gamma^L \cdot L(t-\Delta t) \tag{23}$$

[4] In practice, the range of activities was $[0,1]$ and we used $nc = 0.1$

The forgetting factor γ^L modulates the time period during which no unexpected corrections must occur before the attractor shaping end. The prediction $\hat{\mathbf{C}}$ of the corrections is learned by conditioning with $\mathbf{C}$ the unconditional stimulus and $\mathbf{R}^{VM}$ the conditional stimulus (24).

$$\hat{C}_m = \sum_i W^C_{mi} \cdot R^{VM}_i \text{ with } \Delta W^C_{mi_r} = \varepsilon^C \cdot R^{VM}_{i_r} \cdot (C_m - \hat{C}_m) \tag{24}$$

The learning rate ε^C is small to have a memory effect. The learned muscle activations are expected to maintain the arm close to the postural target, so no more corrections are necessary. The learning of this posture can then stop and the motor babbling resumes. Sometimes, the arm may be blocked by an obstacle (possibly itself). The current version of the architecture does not include an obstacle avoidance process (still, a security module can block movements to prevent damages), so the muscles may only be more and more contracted without correcting the position. The deadlock is broken when the prediction $\hat{\mathbf{C}}$ of the continuous correction finally compensates the detected correction C and stops the learning. The motor babbling can then resume and the muscle activations related to this unsuccessfully learned postural attractor, is not used for the control. Interestingly, in [44], the authors hypothesized that the role of dopamine could also be to detect novelty and maintain or repeat recent actions (not only to reinforce learning). In our case, detecting unpredicted situations (corrections) can maintain the learning of a given posture instead of resuming the motor babbling.

5 Experimental results

5.1 Postural attractor learning

The process to learn postural attractors was tested and validated in a simulation[5] of the Katana arm used in our robotics

[5] with the software Webots (Cyberbotics)

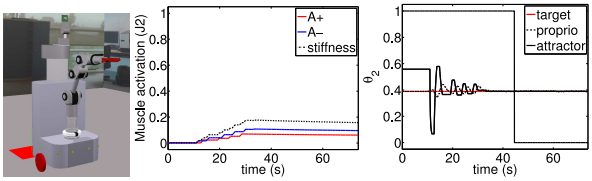


Fig. 7 Webots simulation of a Katana arm. Learning a postural attractor in the 4 DOF motor space. The evolution of the muscle activation and of the resulting equilibrium point is given for the 2^{nd} articulation of the arm. A uniform random noise ($[-0.5, 0.5]$) is added to the torque command. When the movement of a joint is in the direction opposite to the target direction, the corresponding muscle activation is increased. As the stiffness increases, the shift of the position of the equilibrium point at each correction becomes smaller enabling to perform a gradient descent toward the target position. Besides, a bigger stiffness increases the robustness to the noise.

experiments (Fig. 7 and 8). As the arm moves, the muscle activations are increased so that each joint is maintained at the desired position (Fig. 7). The progressive adaptation of the muscle activations depends on random movements (12). Still, the arm finally stabilizes at the desired posture (Fig. 8). As the muscle activations increase, the shifts of the equilibrium point due to learning become smaller and smaller. This property results from the ratio in the equation of the equilibrium point ($\theta_{j,eq} = \frac{A_j^+}{A_j^+ + A_j^-}$). So, the equilibrium position converges to the desired position while the stiffness ($K_j = A_j^+ + A_j^-$) increases. The behavior adaptation is quite slow because of the low frequency of the hardware control loop of the Katana arm (about 7 Hz). Another major constraint is the speed encoding in the robot arm firmware. Very low speed is not available because of the discretization of the values. Instead of an unnatural freezing of movements when the speed should be very close to null, the articulations keep rotating at the fixed minimal speed. These small oscillations give in fact a more natural aspect to the idle movements of the arm. The feeling of a frozen system is avoided during human robot interaction.

5.2 DM-PerAc visuomotor controller

We validated the visuomotor controller in the same 3D simulation of a Katana robot arm as in previous section. In Figure 9(a-c), the robot performs a motor babbling with parameters inducing a low selectivity and thus a very low level of accuracy for the recruited visual and proprioceptive states. A simple test to evaluate the visuomotor learning is to reproduce a trajectory given in the visual space. A star shaped trajectory is given as visual input to the system (Fig. 9b). The trajectory resulting from the visual processing of the arm end effector tracking is displayed. The robot tries to follow the trajectory but because of its sparse learning, the performance is very limited. In the developmental process of the robot, the parameters determining the sparsity of learning may be changed to recruit more visual and proprioceptive categories (Fig. 9(d-f)). The new visuomotor attractors are integrated on-line to the initial learning. The performance of the system is increased. Figure 10 displays the visual trajectories of the desired and real position of the arm end effector. The mean square error is shown with the mean error and the standard deviation to compare the evolution of the performance with the inclusion of more attractors. The same kind of performance could have been obtained by directly learning with the parameters increasing the selectivity of the coding. However, we consider that a progressive refining of the selectivity is important to let the robot learn rapidly how to act. Learning a postural attractor takes time, learning many attractors will slow the exploration of the whole motor space. After a fast learning with the DM-PerAc model, the robot can perform simple tasks even if with limited accuracy. At any time, the DM-PerAc model can let learning

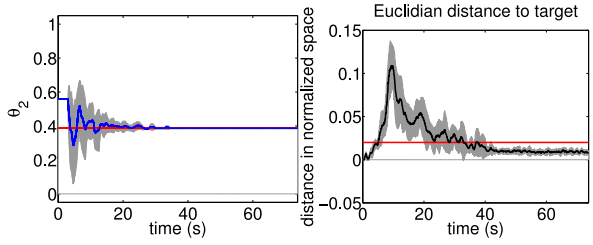


Fig. 8 The attractor learning test is reproduced 10 times. Left: Mean position of the learned attractor for joint 2 with the limits of the gray area representing the standard deviation. Right: Average and standard deviation of Euclidean target distance in the normalized joint space. The red line is the distance constraint th_D for each joint proprioception. The mean distance to the target decreases down to this constraint.

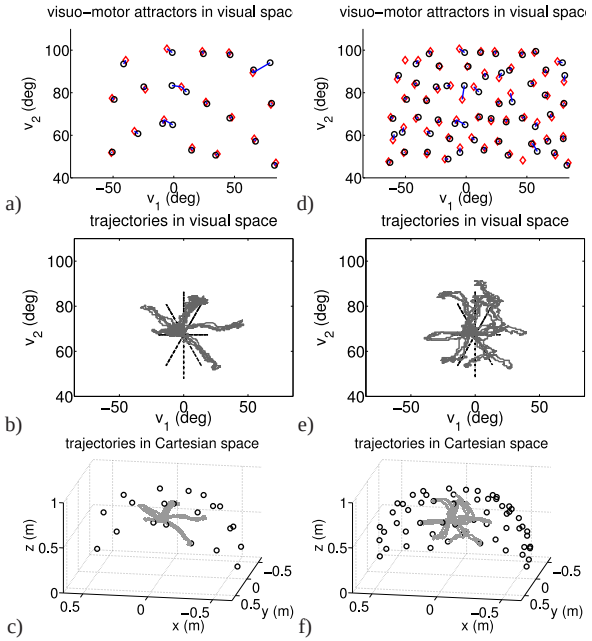


Fig. 9 Simulation of on line learning and adaptation of sensorimotor attractors with a 4 DOF arm and a 2D camera. Left hand column presents the results after an initial sparse learning and the right hand column gives the results after learning continued with learning parameters inducing more selectivity in the state recruitment. *a)* During the motor babbling, the robot recruits visual states (red diamonds) and proprioceptive states (black circles). Each proprioceptive state is associated with one visual state (blue link). *b)* After learning, the visual input is artificially switched to a star shaped trajectory in the visual space (dark line). According to the visual state recognition, the robot moves so the arm end effector trajectory tries to follow the visual input (gray dashed-line). *c)* Movements performed in the 3D Cartesian space during the star shaped trajectory reproduction. *d)* As the parameters changes, the robot can complete its previous learning by recruiting more visual states and proprioceptive states. *e)* The movements of the arm matches more closely to the star shaped trajectory in the visual space. *f)* Corresponding movements in the 3D Cartesian space.

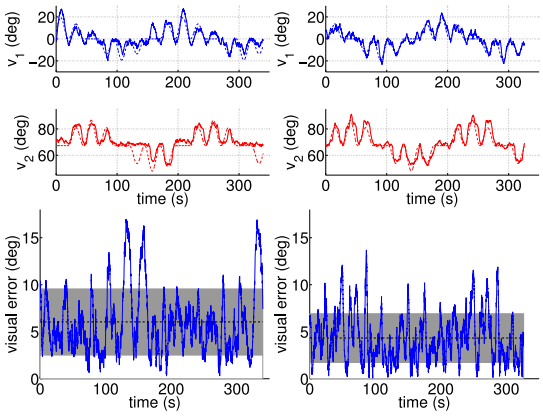


Fig. 10 Comparison between the trajectories from initial (Left hand column) and consecutive learning (Right hand column). Initial learning: mean error 6.0 degrees, standard deviation 3.5 degrees. Consecutive learning: mean error 4.3 degrees, standard deviation 2.5 degrees.

continue in order to complete the initial learning, improving the capabilities of the robot. New visual and proprioceptive categories can be recruited while the motor babbling is resumed. Therefore, the robot must continuously evaluate the co-occurring proprioceptive and visual inputs to improve its visuoarticular model with the newly learned categories. The visuoarticular associations must be progressively updated as the system continues its babbling. Hence, obtaining a visuomotor control with a good level of accuracy can be quite long. Such a learning time is consistent with the development of visuomotor reaching behaviors in infants.

In [14], several regression algorithms (including LWPR [55]) were compared on the visuomotor control learning and performance. The evaluation task is target tracking by the arm end effector of a robot. A stereo camera detects the target, and its 3D Cartesian position is computed. In most of the tests, the target follows a star shaped trajectory path in a vertical plane. The regression algorithms learn a forward kinematic model in order to perform the tracking, thus focusing the exploration process on the motor space to perform the task. The forward model allows to estimate the Jacobian matrix of the kinematic model, the inversion of this matrix and the 3D position of the target provide the motor control of the robotic arm. In this article, we have tested the DM-PerAc visuomotor controller on tracking a target moving on a star shaped trajectory. In our experiment protocol, the visuomotor learning is open-ended. Also, the target coordinates are simulated (no occlusion) in the 2D visual space. The trajectories after learning are comparable to those obtained in [14]. Still, the regression techniques produce smoother trajectories more accurate at the points of the star path. However, inverting the Jacobian matrix requires a specific processing in order to avoid singularities. Such a matrix inversion is not satisfying in the perspective of the developmental approach and is also difficult to model as a biologically plausible process.

5.3 Bifurcation property of the DM-PerAc controller

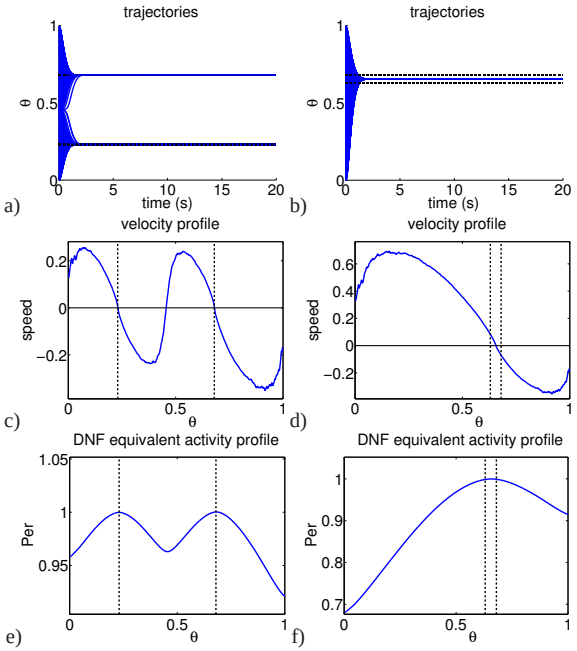


Fig. 11 Bifurcation capabilities in the DM-PerAc controller. Top row (a-b) shows the trajectories (blue lines) and the two learned attractors (black dashed lines). The middle row (c-d) displays the angular velocity profiles in function of the proprioception θ. The bottom row (e-d) gives the perception activity profile equivalent to the activities in a Dynamic Neural Field. In left hand column, the learned attractors are distinct whereas in the right hand one they are closer resulting in one merged behavioral attractor.

We compare the properties of the DM-PerAc controller with the properties of the Dynamic Neural Field based controller. Dynamic Neural Fields (DNF) based on Amari equation [2] are a solution to motor control used to navigate [52], [25] or to control a robotic arm [33, 3]. Biological studies showed that the activations of some neurons in the motor cortex are correlated with the direction of the movement to be performed [23]. In DNF, the activity profile of the field takes the shape of a Gaussian centered on the input stimuli. Besides, the derivative of the activity profile can provide the dynamics of the control [52]. Dynamic Neural Fields have interesting dynamical properties: memory to filter non stable or noisy stimuli and bifurcation capabilities enabling reliable and coherent decision when multiple stimuli are presented.

In Figure 11, we show that (i) the trajectories generated by the DM-PerAc model can be analyzed and integrated to build the DNF equivalent profile of activity, and (ii) there

are bifurcation capabilities in our controller. In our tests, the state space is $[0,1]$. Trajectories generated by the DM-PerAc controller are averaged into the actions $Ac(\theta)$ depending on the state of the system (position). In practice, $Ac(\theta)$ is discretized into a vector with components that are the values for different θ. The result is thus the velocity profile given in Fig 11c and d. In [41], we proposed that the action *Ac* is the derivative of a potential function defining the perception of the system. The action *Ac* is thus spatially integrated to obtain the perception *Per* (25).

$$\forall k, Per_k = \int_{[0,k/n]} Ac(\theta) d\theta + cste \quad (25)$$

where *Per* is a vector of dimension n with components equal to the integration of the action *Ac* at different positions $\theta = k/n$. The integration constant *cst* is chosen so the maximal component value of *Per* is equal to 1. The perception profile *Per* is equivalent to the activity profile of a DNF, and shows bifurcation properties (see Fig 11). The DM-PerAc model can produce behaviors similar to those obtained with the use of an explicit DNF without the need to define the whole field. However, the property of memory is not directly available in the model, but some others processes could complete the DM-PerAc architecture to obtain this property.

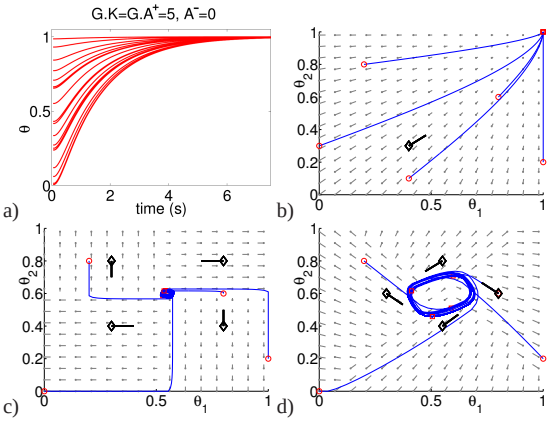


Fig. 12 *a)* Trajectories in 1D space with an asymetric muscle activation pattern (a muscle is inactive). Trajectories start from different random positions. Activation signals are $G \cdot A^+ = G \cdot K = 5$, $A^- = 0$. The control parameters are $\sigma = 5$, $\Delta t = 0.05$ and the moment of inertia $I = 1$. *b-c)* Attraction basins in a bounded 2D space $[0,1]^2$ with DM-PerAc model. Given the learned position/movement couples (black diamonds, thick black lines), a force field is generated (small gray points and lines). For each joint, only one of the agonist/antagonist muscles is activated like in a). Initial (circle) and final (square) points of the trajectories are indicated. *b)* Vector field corresponding to one learned proprioception/activation couple. *c)d)* Four state/action couples are learned. Four trajectories with different starting points are represented in the 2D state space. With only four couples, the system can learn a loop trajectory. The size of the loop depends on the speed thus is related to the damping factor σ and the stiffness K. *c)* $\sigma = 10, G \cdot K = 10$. *d)* $\sigma = 5, G \cdot K = 10$. The other parameters of the system are the time increment $\Delta t = 0.05$ and the moment of inertia $I = 1$.

6 Use and refinement of the DM-PerAc model

6.1 Encoding trajectories with the DM-PerAc controller

The DM-PerAc architecture is not limited to defining fixed point attractors. Now, we consider the case where only one of the muscles around a joint is activated (activation different of 0) while the other one is inactive. This configuration of activation signals induces movement toward the extreme limit of joint (full flexion or extension) (Fig. 12a). At the lower level of motor control, the muscle activations can generate a stable attractor point or encode the direction of movement.

Associations between sensory categories and such pattern of activation can be used to shape trajectories by learning sensorimotor couples. The studied task is simply to reproduce a loop in the 2D motor space. Among four encoded states, each of them are associated with two 1D controllers i.e. four muscle activation coefficients each. The muscle activations correspond to the demonstrated direction of movement. For each joint, only one of the muscle activation (agonist or antagonist) is different of null. An example of a vector field in 2D space defined by one state/action couple is given in Fig. 12b. An attraction basin can effectively be generated (Fig. 12c and d). The trajectories in the 2D state space show that the stiffness K and the damping factor σ control the movement speed and thus can change the size of the loop. Trajectories could be encoded using the low level state/muscle activation associations. This coding can thus be a basis for both posture and trajectory encoding. In the next section, we will focus on learning stable postural attractors.

6.2 Imitative behaviors with the DM-PerAc controller

The visuomotor controller based on the DM-PerAc model can be used for the emergence of low level imitative behaviors and can even be a basis for deferred imitation. An arm controller, based on learning visuoarticular associations, can let low level imitation emerge [3]. In a first phase of babbling, the robot learns its body schema as multiple associations between the visual position of its arm end effector and the joint configuration of its arm. If the robot visual perception is enough limited (using only movement information or the detection of colored patches), the robot can look at the hand of an interacting human and still believes it is its own hand. According to the previously learned visuoarticular associations, this situation can induce an incoherence between the visual information from the teacher's hand and the motor information from the hand of the robot. As the controller is a homeostat, it tends to maintain equilibrium between the visual and the motor signals. Thus, the robot tries to reduce the visuoarticular incoherence by moving its

hand to match the visual input. Low level imitation emerges as the movements of the robot follow the movements of the human (Fig. 13). In the next stage of development of the

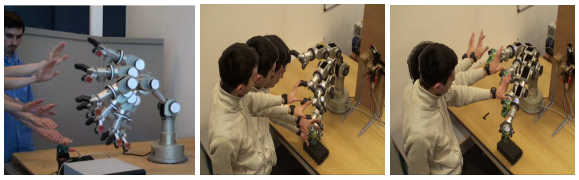


Fig. 13 Example of imitation behaviors. *Left* : Low level imitation of meaningless gestures. Qualitative comparison of imitated gestures performed in front of the robot. Perception ambiguity and a homeostatic controller induce movements to maintain perceptual equilibrium. The robot performs low level imitation of directly observed gestures. *Middle and right* : Gesture imitation can be used to bring the arm end-effector toward objects (here, to grasp a can) or interesting parts of the environment. It can become a common basis for learning by observation and learning by doing.

robot, this low level visuomotor controller can be the basis for learning from observation. We consider that the learning robot can now memorize the sequence of the visual positions demonstrated by the teacher while it is inhibiting its own movement [45]. Then, as the robot internally rehearses the encoded visual sequence, the predicted visual position of the next state can be given to the low level visuomotor controller. The robot reproduces the demonstrated sequence of gestures according to what was perceived during the demonstration. The robot is capable of doing some deferred imitation [45, 46].

6.3 Attractor selection and visuomotor control refining

The refining potential of the DM-PerAc model can be enhanced by the "Yuragi" (fluctuations) based attractor selection [18] which relies on the following Langevin's equation:

$$\Lambda \cdot \dot{\mathbf{x}} = \xi \cdot f(\mathbf{x}) + \eta$$

where Λ is a time constant, the vector $\mathbf{x}$ describes the state of the system and the function f generates an attraction basin resulting from the combination of known attractors. Feedback on the current movement performance modulates the coefficient ξ of the attractor function versus a stochastic exploration term η. As a result, the system can switch from exploration between the different known attractors to exploitation of the closest attractor. In Figure 14, we tested the reaching of a visual position using the "Yuragi" principle [45]. The robot arm end effector reaches the visual target both when it is near the position of a learned attractor (Fig. 14(a-b)) and when it is between the learned attractors (Fig. 14(c-d)). While performing tasks, the robot can use the "Yuragi" principle to reach targets which were not explicitly learned as attractors. When necessary, a new attractor could be recruited to learn how to reach a target that would otherwise be long to reach. The performance of the visuomotor controller could be improved for particular cases without recruiting many useless attractors.

7 Conclusion-discussion

Our previous works enabled to explain trajectory learning (PerAc model [21]) and imitative behaviors [3]. Even though these different works have in common the sensorimotor learning principle, their properties could not directly be combined due to motor control issues. We propose the Dynamic Muscle PerAc (DM-PerAc) model to control a robot arm with multiple DOF (Sec. 4). It combines the principles of the PerAc model with a simple model of agonist/antagonist muscles where the muscle activations determine the movements of the robotic arm. The low level motor control is equivalent to impedance control. The DM-PerAc model can incrementally learn on-line the visuomotor control of the robot arm. During a motor babbling process, proprioceptive and visual categories are recruited and associated together (kinematic model) depending on co-activation. The DM-PerAc model then learns the postural attractors associated with the visuoarticular categories to define the visuomotor control. Trajectories can also be coded by combining state/action couples like in the PerAc model (Sec. 6.1). The states are associated with asymmetric muscle activations to generate movements in particular directions. In section 6.2, we showed that imitative behaviors can be obtained with the DM-PerAc visuomotor controller. This controller can also be a basis for higher level encoding and imitation behaviors. Finally, the "Yuragi" principle may be used to increase accuracy without learning more sensory categories (Sec.6.3).

In this article, the motor control is based on a spring based model of muscles ; however, we do not pretend that modifying the stiffness of these spring-like muscles corresponds to an accurate model of neuro-muscular control. The rest-length of the muscles, motor reflexes and other physiological properties are also important. Still, the aim of the DM-PerAc model is to allow sensorimotor attractor learning with the attractors that can be either postures or trajectories. Using muscle activations has the advantage to make learning easier whatever the attractor is (posture or trajectory).

The computational cost of the DM-PerAc visuomotor controller can be reduced in different ways. The neurons corresponding to categories (visual, proprioceptive, visuoarticular) not yet recruited can be ignored in the neural update process. Also, the number of visual to visuoarticular links (W^{VM}) may be reduced by using some lists of links dynamically managed according to the recognition of the visual and proprioceptive categories. This solution would allow to use

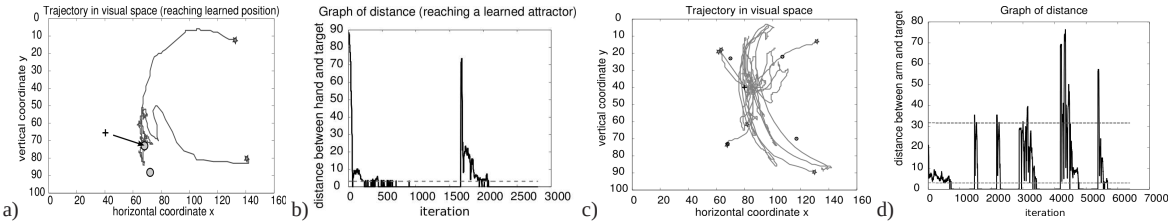


Fig. 14 Visual target reaching with a visuomotor controller using the "Yuragi" principle. The feedback is based on the target distance in the visual space. A known attractor can match the target (a,b) or the target can be between learned attractors (c,d). *(a,c)* Trajectories of the robot arm end effector in the visual space. The black circles correspond to the learned attractors and the black cross is the visual target to be reached. The stars are the starting positions for each trial. *(b,d)* Evolution of the distance between the arm end effector and the target in the visual space (number of pixels). Dark gray dash-line shows the average distance to the attractors. The light gray line shows the threshold under which the target is reached. *a*: Trajectories while reaching a learned attractor, 2 attractors activated, 2 trials with different starting positions. *b*: Corresponding evolution of the target distance. *c*: Trajectories while reaching a not previously learned position, 4 attractors activated, 6 trials with different starting positions. *d* Corresponding evolution of the target distance. In both cases, the arm end effector reaches the target, although, when it is not a learned position, the reaching can be quite long due to the random exploration.

far fewer links than if considering the whole set of visual to visuoarticular links.

We gave solutions to learn attractor points as they are used in the visuomotor controller for imitation behaviors. The learning of trajectories or paths is not described in this article. In the DM-PerAc model, postural attractors can be used as via-points to encode trajectories and we used this kind of solution in deferred imitation [45]. However, a correct encoding of dynamic trajectories should rely on state/action couples defining attraction basins, like in the PerAc model (Sec. 3.1). The advantage is that agonist and antagonist muscles would not need to be active at the same time. The stiffness and the energy consumption can be reduced. Besides, this trajectory encoding is coherent with the observed tri-phasic pattern of movement in which agonist and antagonist muscles are alternatively active [48]. A state-action coding may allow the tri-phasic pattern though anticipatory capabilities seems necessary in the model to exactly obtain the different phases of activation.

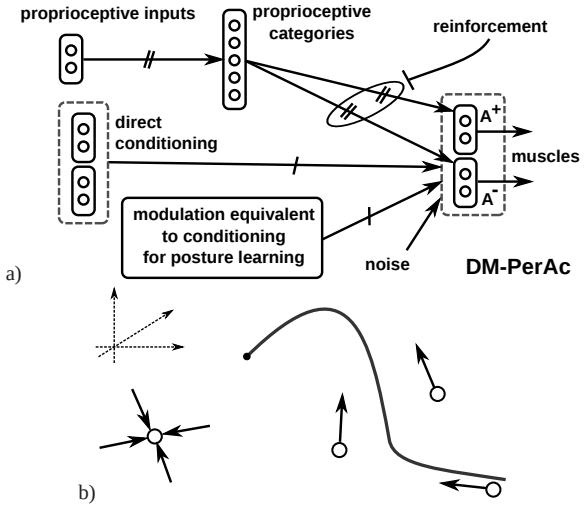


Fig. 15 *a)* Possible solutions to learn muscle activations in the Dynamic Muscle PerAc model. b) Example of dynamic trajectory with postural attractors and trajectory shaping constraints. Both components can be coded similarly in the DM-PerAc architecture.

Although we proved that the DM-PerAc model enables dynamical trajectory encoding, the learning of the adequate state/action couples is still an ongoing issue. In the PerAc model, the states and actions were associated by direct conditioning. The orientation to follow (action) could be estimated by integrating the followed orientation while moving. The orientation to follow could also be demonstrated to a passive robot. In the DM-PerAc model (Fig 15a), a direct conditioning is possible but a particular process is necessary to extract the unconditional stimulus from a passive demonstration. Changes of proprioception cannot be directly converted into muscle activations (for instance, the muscle activations must change to perform the same movement manipulating objects with different masses). The "Yuragi" process, adapted in the DM-PerAc model, can be a potential solution to this issue. In the "Yuragi" controller [18], the attractor function f is based on a combination of positional attractors whose weights depend on the distance between the current and the encoded positions. Learning corresponds to shifting the encoded positions of some attractors. In the DM-PerAc model, these two principles are modified. The attractors that are combined are the 1 dimensional attractors corresponding to the muscles performing flexion or extension of the joints. These attractors can encode either a direction of movement or a position. The combination is learned by storing the weights of these attractors as muscle activations. The postural learning described in Section 4.2 illustrates these different adaptations of the "Yuragi" process. Given the adequate feedback, the combination of attractors is learned to maintain a posture. This learning relies on the reinforcement of the muscle activation but with respect to a constraint given by supervision. The learning thus shares similarities with both reinforcement learning and conditioning.

We believe that the same learning process could be adapted to learn combinations of attractors to fulfill other kinds of constraint like following particular speed vectors. This is possible because the 1D attractors can also code direction of movement. Still, the remaining issues are what the adequate feedback is and how it can be learned from a demonstration. Finally, using the same encoding and the same kind of learning, the robot should be able to learn trajectories like in Fig.15b mixing posture attractors and trajectory shaping.

Appendices

A Summary of the parameters and equations used in the Dynamic-Muscle PerAc model

The different parameters and equations presented in this article are respectively summarized in Table 1 and Table 2.

The proprioceptive (visual) categorization depends on the vigilance parameters λ^P (λ^V) and the parameters β^P (β^V) of the Gaussian similarity measure. High vigilance values would imply that recruited categories overlap. We use $\lambda^P = \lambda^V = 0.005$ to avoid interferences between categories. The values of the Gaussian parameters are very low so the categories are selective enough. During the learning step, different values are used to increase progressively the number of learned categories ($\beta^P = 0.002$ then $\beta^P = 0.001$, and $\beta^V = 2 \cdot 10^{-4}$ then $\beta^V = 5 \cdot 10^{-5}$). During the tests, the vision must drives the movements, so the proprioceptive categories must be less selective than the visual categories ($\beta^P = 0.1$ and $\beta^V = 5 \cdot 10^{-5}$).
In the experiments, the muscle activation learning depends on the learning factor $\varepsilon^A = 10^{-3}$ and the decay factor $\alpha^A = 10^{-4}$. As the learning factor is small, the stiffness K_j of each joint changes slowly. Still, the equilibrium position is rapidly adapted because it depends on the ratio of the muscle activations. Also, the decay must be slow enough to allow the learning. With an error threshold $th_D = 0.01$, the muscle activations around a joint are adapted if the position error is over a hundredth of the rotational range.
The parameters th_L and γ^L define the dynamics of the "learning enable" signal L i.e. determine the amount of time to learn each postural attractor. The used values are $th_L = 10^{-5}$ and $\gamma^L = 0.95$ so the motor babbling resumes after a time period of about 10 seconds without correction of the movements.

Acknowledgements This work was supported by the INTERACT french project referenced ANR-09-CORD-014.

References

1. Albu-Schäffer A, Ott C, Hirzinger G (2007) A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. Int J Robot Res 26(1):23–39
2. Amari SI (1977) Dynamics of pattern formation in lateral-inhibition type neural fields. Biol Cybern 27(2):77–87
3. Andry P, Gaussier P, Nadel J, Hirsbrunner B (2004) Learning invariant sensorimotor behaviors: a developmental approach to imitation mechanisms. Adapt Behav 12(2):117–140
4. Argall BD, Chernova S, Veloso M, Browning B (2009) A survey of robot learning from demonstration. Robot Auton Syst 57(5):469–483
5. Atkeson CG, Andrew W, Stefan Schaal Z (1997) Locally weighted learning. In: Artif. Intell. Rev., pp 11–73

Table 2 DM-PerAc Model: equation summary

Motor control based on commands of stiffness of agonist/antagonist muscles around the joints j[a]

$$\tau_j = A_j^+ \cdot \theta_j^+ - \sigma_j^+ \cdot \dot{\theta}_j^+ - (A_j^- \cdot \theta_j^- - \sigma_j^- \cdot \dot{\theta}_j^-)$$

Which is simplified from additional constraints (11) as:

$$\ddot{\theta}_j = \frac{K_j}{I_j} \cdot (\theta_{j,eq} - \theta_j) - \frac{\sigma_j}{I_j} \cdot \dot{\theta}_j \text{ with } K_j = A_j^+ + A_j^- \text{ and } \theta_{j,eq} = \frac{A_j^+}{A_j^+ + A_j^-}$$

Update and learning of the proprioceptive and visual categories
Proprioceptive categories (index i) based on the muscle proprioception $\mathbf{P} = [\theta_1^+, \theta_2^+ \ldots, \theta_1^-, \theta_2^-, \ldots]$ (index m):

$$S_i^P = exp(-\frac{\sum_m (P_m - W_{im}^P)^2}{2\beta^P})$$
$$\Delta W_{i_r m}^P = \varepsilon^P \cdot (P_m - W_{i_r m}^P) \text{ with } \varepsilon^P = \mathscr{H}(\lambda^P - \max_i(S_i^P))$$

Visual categories (index k):

$$R_k^V = \frac{S_k^V}{\sum S^V} \text{ with } S_k^V = exp(-\frac{\sum_l (V_l - W_{kl}^V)^2}{2\beta^V})$$
$$\Delta W_{k_r l}^V = \varepsilon^V \cdot (V_l - W_{k_r l}^V) \text{ with } \varepsilon^V = \mathscr{H}(\lambda^V - \max_k(S_k^V))$$

Visuoarticular association learning

$$\Delta W_{ik}^{VM} = \varepsilon^{VM} \cdot S_i^P \cdot (f(S_i^P) \cdot f(R_k^V) - W_{ik}^{VM})$$
$$\text{with } f(X_l) = 1 \text{ if } X_l = \max_l(X_l) \text{ and 0 otherwise}$$

Visuoarticular categories update

$$\begin{cases} R_i^{VM} = \frac{S_i^{VM}}{\sum S^{VM}} \quad \text{with} \quad S_i^{VM} = R_i^P \cdot \sum_k (g(W_{ik}^{VM}) \cdot R_k^V) \\ \text{and } g(W_{ik}^{VM}) = 1 \text{ if } \left(\frac{W_{ik}^{VM}}{\max_k(W_{ik}^{VM})} \right)^n > 0.5 \text{ and 0 otherwise} \end{cases}$$

Postural attractor learning
Reinforcement signal based on incorrect movements:

$$\begin{cases} C_m = \mathscr{H}(M_m^D - M_m) \text{ where} \\ M_m^D = \mathscr{H}(P_m - \hat{P}_m - th_D) \text{ and } M_m(t) = \mathscr{H}(P_m(t - \Delta t) - P_m(t)) \\ \hat{P}_m = \sum_i W_{mi}^{\hat{P}} \cdot R_i^{VM} \text{ with} \\ W_{mi_r}^{\hat{P}} = \varepsilon^P \cdot (P_m - W_{mi_r}) \text{ (on recruitment of a new } R_{i_r}^{VM}) \end{cases}$$

Signal L indicating whether attractor learning should continue:

$$\begin{cases} L(t) = \mathscr{H}(L(t - \Delta t) - th_L) \cdot \sum_m [C_m - \hat{C}_m]^+ + \gamma^L \cdot L(t - \Delta t) \\ \hat{C}_m = \sum_i W_{mi}^C \cdot R_i^{VM} \text{ with } \Delta W_{mi_r}^C = \varepsilon^C \cdot R_{i_r}^{VM} \cdot (C_m - \hat{C}_m) \end{cases}$$

Muscle activation update and learning:

$$A_m = \sum_i W_{mi}^A \cdot R_i^{VM} \text{ with } \mathbf{A} = [A_1^+, A_2^+, \ldots, A_1^-, A_2^-, \ldots]$$

$$\Delta W_{mi}^A = \mathscr{H}(L - th_L) \cdot (\varepsilon^A \cdot C_m \cdot R_i^{VM} \cdot (1 - W_{mi}^A) - \alpha^A \cdot W_{mi}^A \cdot \max_j [K_j - nc]^+)$$

During attractor learning ($L > th_L$), after the recruitment of the visuoarticular category i_r, a bias on the activation ensures that $R_{i_r}^{VM} = 1$ and $R_{i \neq i_r}^{VM} = 0$.

[a] The time step t is only written when a signal at different time step is used

Table 1 DM-PerAc Model: parameter summary with values used in experiments for the open parameters

$\mathbf{A}$ = $[A_1, \ldots, A_{2N}]$ muscle activation (stiffness)

$\mathbf{A}^+$, $\mathbf{A}^-$: activation of agonist ($^+$) and antagonist ($^+$) muscles for each joint ($\mathbf{A} = [\mathbf{A}^+, \mathbf{A}^-]$)

$\mathbf{C}$: comparison of desired and current movements, determines the need to correct muscle activations, modulates the increase of $\mathbf{W}^A_{mi}$

$\hat{\mathbf{C}}$: prediction of $\mathbf{C}$ for a given visuoarticular category i in $\mathbf{R}^{VM}$

G : stiffness factor, counterbalancing the bounded muscle activations $\mathbf{A}$ (ex: $G = 60$)

$\mathbf{K}$: stiffness

i, i_m, i_r : indexes of proprioceptive category, winning proprioceptive category and next recruited proprioceptive category

$\mathbf{I}$: moment of inertia (ex: $I = 1$)

j : index of joint

k, k_m, k_r : indexes of visual category, winning visual category and next recruited visual category

l : visual coordinates

L : attractor learning signal

m : index of muscle

$\mathbf{M}^D$, $\mathbf{M}$: desired muscle shortening, current muscle shortening

n : exponent, used in the update of the visuoarticular categories (ex: $n = 100$)

N : number of joints

$\mathbf{R}^P$, $\mathbf{R}^V$,$\mathbf{R}^{VM}$: normalized activities of $\mathbf{S}^P$,$\mathbf{S}^V$ and $\mathbf{S}^{VM}$

$\mathbf{P}$ = $[P_1 \ldots P_{2N}] = [\mathbf{P}^+\mathbf{P}^-]$ proprioceptive input

$\mathbf{P}^+$, $\mathbf{P}^-$: agonist and antagonist proprioceptive input $[\theta_1^+\theta_2^+\ldots]$, $[\theta_1^-\theta_2^-\ldots]$

$\mathbf{S}^P$, $\mathbf{S}^V$: recognition activities of proprioceptive and visual categories respectively

$\mathbf{S}^{VM}$: visuo-articular category, merging visual and proprioceptive signals

t, $t-\Delta t$: current time step, previous time step

th_D : threshold on target distance to estimate desired movement (ex: $th_D = 0.01$)

th_L : threshold on L under which current attractor learning is stopped (ex: $th_L = 10^{-5}$)

$\mathbf{V}$: visual input (coordinates in visual field)

$\mathbf{W}^P_{im}$, $\mathbf{W}^V_{kl}$: learning weights to proprioceptive ($\mathbf{S}^P$) or visual ($\mathbf{S}^V$) categories

$\mathbf{W}^C_{mi}$: learning weights to $\hat{\mathbf{C}}$

$\mathbf{W}^A_{mi}$: learning weights to $\mathbf{A}$

$\mathbf{W}^{VM}_{ik}$: learning weights to $\mathbf{R}^{VM}$

α^A : decay factor of muscle activation learning ($\mathbf{W}^A_{mi}$) (ex: $\alpha^A = 10^{-4}$)

β^P, β^V : variance parameter of the Gaussian kernels of proprioceptive P or visual V categories.

ε^A : learning factor of muscle activation ($\mathbf{A}$) learning (ex: $\varepsilon^A = 10^{-3}$)

ε^C : learning factor of the predictor of $\mathbf{C}$ (ex: $\varepsilon^C = 0.2$)

ε^P, ε^V : learning factor of proprioceptive P or visual V categorizations.

γ^L : forgetting factor of the attractor learning signal L (ex: $\gamma^L = 0.95$)

λ^P, λ^V : vigilance of proprioceptive categorization P or visual categorization V. (ex: $\lambda^P = \lambda^V = 0.05$)

σ_j : damping factor (ex: $\sigma_j = 11$)

θ_j, $\dot{\theta}_j$, $\ddot{\theta}_j$: rotation angle of a joint, velocity, acceleration

θ_j^+, θ_j^- : positive angular value measured in the agonist or antagonist reference (see Fig. 4)

$\theta_{j,max}$, $\theta_{j,min}$: maximal and minimal angular value of a joint

$\theta_{j,eq}$: equilibrium point resulting from muscle activations

τ_j : rotational torque

General tools

Heaviside function: $\mathscr{H}(x) = 1$ if $x > 0$, 0 otherwise

Kronecker symbol: $\delta_{ij} = 1$ if $i = j$, 0 otherwise

$[x]^+ = x$ if $x > 0$, 0 otherwise

6. Bullock D, Grossberg S (1989) VITE and FLETE: neural modules for trajectory formation and postural control. In: Hershberger W (ed) Volitional action, Adv. Psychol., vol 62, Elsevier, chap 11, pp 253–297
7. Burdet E, Tee KP, Mareels I, Milner TE, Chew CM, Franklin DW, Osu R, Kawato M (2006) Stability and motor adaptation in human arm movements. Biol Cybern 94(1):20–32
8. Calinon S, Guenter F, Billard A (2007) On learning, representing and generalizing a task in a humanoid robot. IEEE Trans Syst, Man, Cybern B Special issue on robot learning by observation, demonstration and imitation 37(2):286–298
9. Calinon S, D'halluin F, Caldwell DG, Billard A (2009) Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. In: Proc. of 2009 IEEE Int. Conf. on Humanoid Robots, pp 582–588
10. Calinon S, D'halluin F, Sauser E, Caldwell D, Billard A (2010) Learning and reproduction of gestures by imitation: an approach based on Hidden Markov Model and Gaussian Mixture Regression. IEEE Robot Autom Mag 17(2):44–54
11. Calinon S, Sardellitti I, Caldwell DG (2010) Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Taipei, Taiwan, pp 249–254

12. Carpenter GA, Grossberg S (2002) Adaptive resonance theory (ART). In: The handbook of brain theory and neural networks, MIT Press, Cambridge, MA, USA, pp 79–82
13. Chiaverini S, Siciliano B, Villani L (1999) A survey of robot interaction control schemes with experimental comparison. IEEE/ASME Trans Mechatronics 4(3):273–285
14. Droniou A, Ivaldi S, Padois V, Sigaud O (2012) Autonomous online learning of velocity kinematics on the iCub: a comparative study. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vilamoura, Portugal, pp 3577–3582
15. Flash T (1987) The control of hand equilibrium trajectories in multi-joint arm movements. Biol Cybern 57(4):257–274
16. Flash T, Hogan N (1985) The coordination of arm movements: an experimentally confirmed mathematical model. J Neurosci 5(7):1688–1703
17. Franklin DW, Burdet E, Tee KP, Osu R, Chew CM, Milner TE, Kawato M (2008) CNS learns stable, accurate, and efficient movements using a simple algorithm. J Neurosci 28(44):11,165–11,173
18. Fukuyori I, Nakamura Y, Matsumoto Y, Ishiguro H (2008) Flexible control mechanism for multi-DOF robotic arm based on biological fluctuation. From Anim Animat 10 pp 22–31
19. G C, L S (1968) The human eye-movement mechanism: experiments, modeling, and model testing. Arch Ophthalmol 79(4):428–436
20. Ganesh G, Albu-Schaffer A, Haruno M, Kawato M, Burdet E (2010) Biomimetic motor behavior for simultaneous adaptation of force, impedance and trajectory in interaction tasks. In: Robotics and Automation ICRA 2010 IEEE International Conference on (2010), IEEE, 2010 IEEE International Conference on Robotics and Automation, ICRA 2010, pp 2705–2711
21. Gaussier P, Zrehen S (1995) PerAc: a neural architecture to control artificial animals. Robot Auton Syst 16(2-4):291–320
22. Gaussier P, Moga S, Banquet JP, Quoy M (1998) From Perception-Action loops to imitation processes: a bottom-up approach of learning by imitation. Appl Artif Intell 1(7):701–727
23. Georgopoulos A, Schwartz A, Kettner R (1986) Neuronal population coding of movement direction. Science 233(4771):1416–1419
24. Giovannangeli C, Gaussier P (2010) Interactive teaching for vision-based mobile robots: a sensory-motor approach. IEEE Trans Syst, Man, Cybern A 40(1):13–28
25. Giovannangeli C, Gaussier P, Désilles G (2006) Robust mapless outdoor vision-based navigation. In: IEEE/RSJ Int. Conf. on Intelligent Robots and systems, IEEE, Beijing, China
26. Hersch M, Billard A (2006) A biologically-inspired model of reaching movements. In: Proc. of the 2006 IEEE/RAS-EMBS Int. Conf. on Biomedical Robotics and Biomechatronics
27. Hill AV (1938) The heat of shortening and the dynamic constants of muscle. Proc R Soc B: Biol Sci 126(843):136–195
28. Hoffmann H, Pastor P, Park DH, Schaal S (2009) biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance. In: Proc. IEEE Int. Conf. Robot. (ICRA2009)
29. Hogan N (1984) An organizing principle for a class of voluntary movements. J Neurosci 4(11):2745–2754
30. Huxley AF (1957) Muscle structure and theories of contraction. Prog Biophys Biop Ch 7:255–318
31. Ijspeert AJ, Nakanishi J, Schaal S (2003) Learning attractor landscapes for learning motor primitives. In: Adv. in neural information processing systems 15, cambridge, ma: mit press, pp 1547–1554
32. Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P, Schaal S (2013) Dynamical movement primitives: learning attractor models for motor behaviors. Neural Comput 25(2):328–73
33. Iossifidis I, Schoner G (2004) Autonomous reaching and obstacle avoidance with the anthropomorphic arm of a robotic assistant using the attractor dynamics approach. In: Proc. IEEE Int. Conf. Robot. (ICRA2004), Inst. fur Neuroinformatik, Ruhr-Univ., Bochum, Germany, IEEE, vol 5, pp 4295–4300
34. Iossifidis I, Schoner G (2006) Dynamical systems approach for the autonomous avoidance of obstacles and joint-limits for an redundant robot arm. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'06), Institut fur Neuroinformatik, Ruhr-Universitat Bochum, pp 580–585
35. Jiménez-Fabián R, Verlinden O (2011) Review of control algorithms for robotic ankle systems in lower-limb orthoses, prostheses, and exoskeletons. Med Eng Phys 34(4):397–408
36. Klute GK, Czerniecki JM, Hannaford B (2002) Artificial muscles: actuators for biorobotic systems. Int J Robot Res 21(4):295–309
37. Kohonen T (1982) Analysis of a simple self-organizing process. Biol Cybern 44(2):135–140
38. Kronander K, Billard A (2012) Online learning of varying stiffness through physical Human-Robot interaction. In: Int Conf on Robotics and Automation, pp 1842–1849
39. Lagarde M, Andry P, Gaussier P, Boucenna S, Hafemeister L (2010) Proprioception and imitation: on the road to agent individuation. In: Sigaud O, Peters J (eds) From Motor Learning to Interaction Learning in Robots, vol 264, Springer Berlin Heidelberg, Berlin, Heidelberg, book part 3, pp 43–63
40. Lungarella M, Metta G, Pfeifer R, Sandini G (2003) Developmental robotics: a survey. Connect Sci 15(4):151–190
41. Maillard M, Gapenne O, Hafemeister L, Gaussier P (2005) Perception as a dynamical sensori-motor attraction basin. In: Capcarrre M, Freitas A, Bentley P, Johnson C, Timmis J (eds) Adv. in Artificial Life, Lecture Notes in Computer Science, vol 3630, Springer Berlin Heidelberg, pp 37–46
42. Miyamoto H, Kawato M (1998) A tennis serve and upswing learning robot based on bi-directional theory. Neural Networks 11(7-8):1331–1344
43. Nehaniv CL, Dautenhahn K (2002) The correspondence problem. In: Dautenhahn K, Nehaniv CL (eds) Imitation in animals and artifacts, MIT Press, Cambridge, MA, USA, pp 41–61
44. Redgrave P, Gurney K (2006) The short-latency dopamine signal: a role in discovering novel actions? Nat Rev Neurosci 7(12):967–975
45. de Rengervé A, Boucenna S, Andry P, Gaussier P (2010) Emergent imitative behavior on a robotic arm based on visuo-motor associative memories. In: IEEE/RSJ Int. Conf. on Intelligent Robots and systems (IROS'10), Taipei, Taiwan, pp 1754–1759
46. de Rengervé A, Andry P, Gaussier P (2013) A brain-based architecture toward action imitation and task learning by observation and demonstration. Front in Neurorobotics p in revision
47. Rozo L, Calinon S, Caldwell D, Jimenez P, Torras C, Jiménez P (2013) Learning collaborative impedance-based robot behaviors. In: AAAI Conf on Artificial Intelligence
48. Sanes JN, Jennings VA (1984) Centrally programmed patterns of muscle activity in voluntary motor behavior of humans. Exp Brain Res 54(1):23–32
49. Schaal S (1997) Learning from demonstration. In: Adv. Neur. In., mit press, vol 9, pp 1040–1046
50. Schaal S (2006) Dynamic movement primitives - a framework for motor control in humans and humanoid robotics. In: Kimura H, Tsuchiya K, Ishiguro A, Witte H (eds) Adaptive Motion of Animals and Machines, Springer Tokyo, pp 261–280
51. Schaal S, Atkeson CG (1998) Constructive incremental learning from only local information. Neural Comput 10(8):2047–2084
52. Schöner G, Dose M, Engels C (1995) Dynamics of behavior: theory and applications for autonomous robot architectures. Robot Auton Syst 16(2-4):213–245
53. Slotine JJE (1988) Adaptive manipulator control: a case study. IEEE Trans Autom Control 33(11):995–1003

54. Todorov E (2007) Optimal control theory. In: Doya K (ed) Bayesian Brain: Probabilistic Approaches to Neural Coding, Applied Mathematical Sciences, MIT Press, chap 12, pp 269–298
55. Vijayakumar S, D'souza A, Schaal S (2005) Incremental online learning in high dimensions. Neural Comput 17(12):2602–2634
56. Winters JM, Stark L (1985) Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models. IEEE Trans Bio-Med Eng 32(10):826–839
57. Winters JM, Stark L (1987) Muscle models: what is gained and what is lost by varying model complexity. Biol Cybern 55(6):403–420

2.2 Propriétés dynamiques du contrôleur : comparaison avec un Champ de Neurones Dynamiques

Nous avons vu dans la section 1.4 que, suivant le modèle PerAc, des associations lieux/actions apprises permettent d'encoder des trajectoires suivies par un robot mobile. Les actions sont alors les orientations à prendre encodées par des champs de neurones dynamiques (*Dynamic Neural Fields*, DNF) [Amari, 1977; Schoner, 1995] (cf. Section 1.3.1). Grâce aux propriétés de bifurcation et d'hystérésis de ces champs de neurones dynamiques, le système a une bonne robustesse au bruit et aux perturbation. Les champs de neurones dynamiques peuvent aussi être utilisés pour contrôler les mouvements d'un bras de robot. Un champ de neurones peut servir à encoder la dynamique pour maintenir une direction de mouvement particulière dans l'espace cartésien 3D [Iossifidis and Schoner, 2004] ou bien être utilisé pour contrôler les mouvements à partir d'informations visuelles [Andry et al., 2004]. La limitation principale des DNF est qu'il faut paramétrer correctement la différence de gaussiennes modélisant les interactions latérales (elles déterminent la distance à partir de laquelle des attracteurs seraient fusionnés ou au contraire séparés). Nous allons étudier l'équivalence entre le contrôleur proposé dans ce chapitre (sec. 2.1) et un contrôleur basé sur des DNF. Pour cela, nous proposons une méthode qui permet d'extraire à partir des trajectoires un champ potentiel que l'on comparera avec le profil d'activité d'un champ de neurones dynamiques. Suivant le contrôle moteur du modèle DM-PerAc (Eq. (14) dans l'article à la section 2.1), nous simulons le contrôle d'une unique articulation à partir de la proprioception θ associée (2.1). Les coefficients d'activation musculaire A^-/A^+ utilisés permettent de générer un attracteur de type point fixe en θ_{eq}.

$$\ddot{\theta} = \frac{K}{I} \cdot (\theta_{eq} - \theta) - \frac{\sigma}{I} \cdot \dot{\theta} \text{ avec } \theta_{eq} = \frac{A^+}{A^- + A^+} \text{ et } K = A^- + A^+ \tag{2.1}$$

Dans le cas que nous allons étudier, l'espace d'état Ω est l'intervalle borné $[0, 1[$ (la variable comportementale θ a donc une seule dimension). Tout d'abord, nous allons calculer le profil dynamique moyen à partir d'un ensemble de trajectoires vers une même cible. Ce profil doit faire correspondre une vitesse de rotation à chaque état proprioceptif P_k. Pour obtenir ce profil, nous devons numériquement intégrer toutes les trajectoires possibles passant par un $\theta \in \Omega$, suivant le principe de la méthode de Monte-Carlo. Le point de départ θ_0 des trajectoires est choisi aléatoirement dans l'intervalle [0, 1[. La distribution de ces valeurs dépend du point attracteur θ_{eq} généré par le contrôle (2.2) :

$$\begin{cases} p(\theta_0 \in [0, \theta_{eq}]) = 1 - \theta_{eq} \\ p(\theta_0 \in [\theta_{eq}, 1[) = \theta_{eq} - 0 \end{cases} \tag{2.2}$$

Pour chacun de ces deux intervalles, un tirage aléatoire suivant une distribution uniforme est utilisé pour obtenir les positions initiales θ_0. Ces précautions sont nécessaires pour éviter le biais dû à l'intervalle borné de l'espace d'état Ω. Autrement, le profil de dynamique obtenu autour du point attracteur devient fortement asymétrique en faveur du sous-intervalle le plus large. L'état du robot est décrit par un vecteur $\mathbf{Sen}$ dont chaque composante correspond à l'une des partitions de l'espace d'état. A chaque instant le système choisi une action qui est réalisée. Ainsi, pour une région P_k de l'espace d'état ($P_k = [(k-1)/n, k/n[$ avec $n = 500$), on extrait l'action moyenne réalisée Ac qui est ici la vitesse moyenne $\dot{\theta}$. Bien entendu, on suppose que la discrétisation de l'espace d'état est suffisante pour que ce moyennage soit cohérent. Dans cette

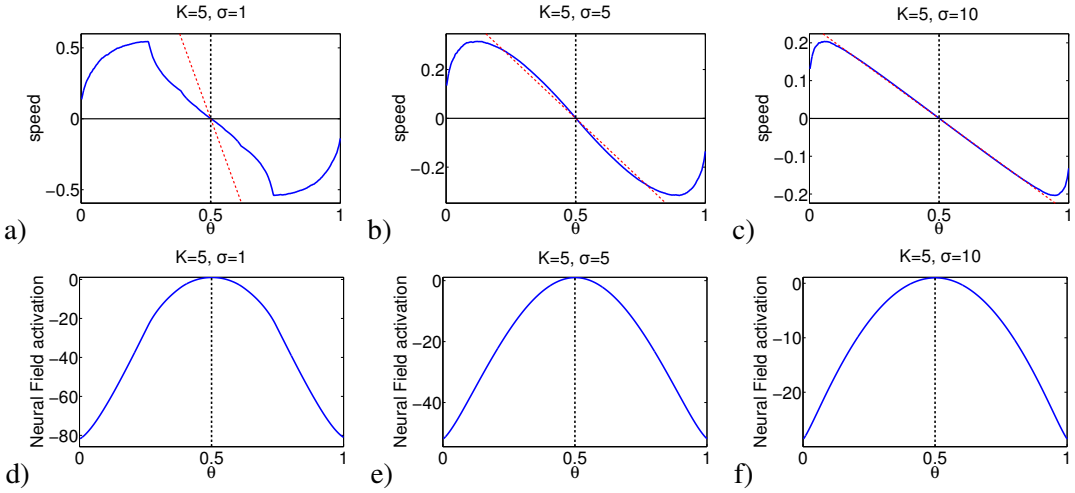


FIGURE 2.2 – Cible en $\theta = 0.5$. Courbes pour des coefficients d'amortissement σ croissant. $\Delta t = 0.1, I = 1$.

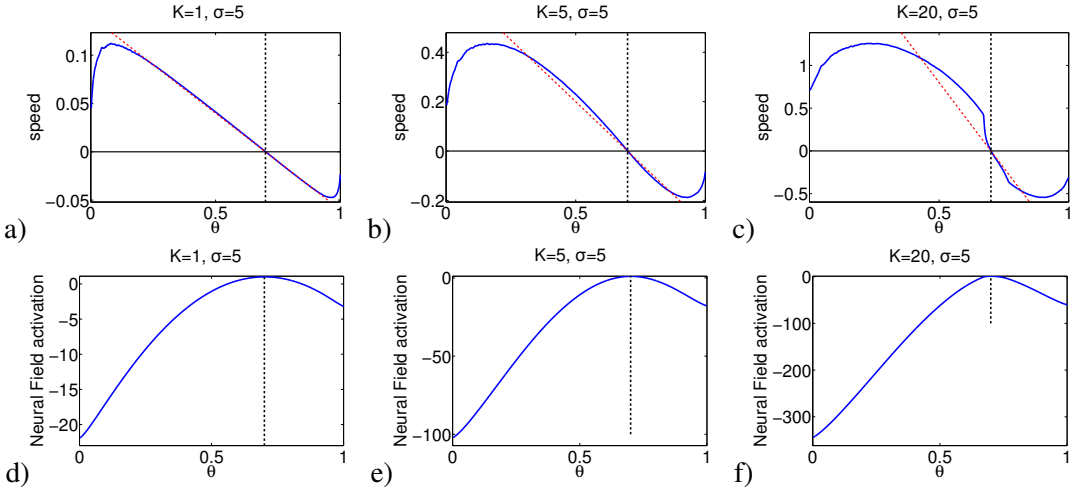


FIGURE 2.3 – Cible en $\theta = 0.7$. Courbes pour des raideurs K croissantes. $\Delta t = 0.1, I = 1$.

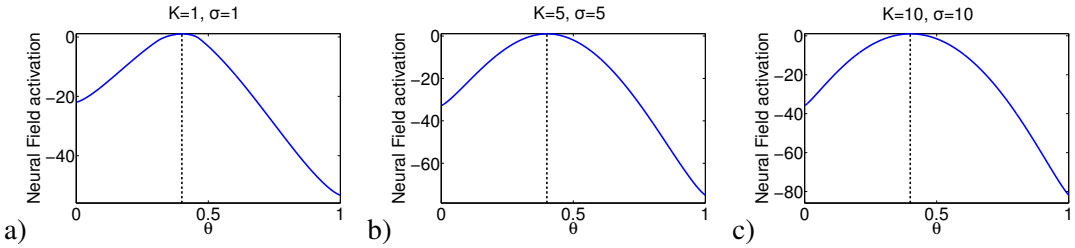


FIGURE 2.4 – Cible en $\theta = 0.4$. Courbe pour une raideur K et une coefficient d'amortissement σ croissant avec un ratio constant. $\Delta t = 0.1, I = 1, ratio = 1$.

situation, la matrice $\mathbf{Ac}.\mathbf{Sen}^\top$ construite décrit le comportement du robot en donnant les actions choisies selon la situation du robot. Pour obtenir $\mathbf{Ac}.\mathbf{Sen}^\top$, il faut intégrer pour chaque région P_k les actions exécutées durant les différentes trajectoires réalisées (2.3).

$$\forall k,\ (Ac.Sen^\top)_k = \begin{cases} \frac{\int_{\mathcal{C}\cap P_k} Ac(r)dr}{\int_{\mathcal{C}\cap P_k} dr} & \text{si } \int_{\mathcal{C}\cap P_k} dr \neq 0 \\ 0 & \text{sinon} \end{cases} \tag{2.3}$$

avec $\mathcal{C}$ l'ensemble des trajectoires possibles. Le vecteur $\mathbf{Ac}.\mathbf{Sen}^\top$ résultant de l'intégration sur les différentes trajectoires est équivalent à un profil de vitesse dépendant de l'état θ (Fig. 2.2(a-c)). Le dénominateur de l'équation (2.3) est nécessaire pour normaliser le profil indépendamment des multiples trajectoires réalisées. L'étude des unités des différents termes nous indique que la normalisation proposée est cohérente avec l'obtention d'un profil de vitesse. D'autre part, nous obtenons des vitesses dont l'amplitude correspond à la prédiction du contrôleur. En effet, l'équation dynamique du contrôleur (2.1) implique que l'accélération devient nulle pour une vitesse critique $\dot{\theta}_c$ (2.4).

$$\dot{\theta}_c = \frac{K}{\sigma}(\theta_{eq} - \theta) \tag{2.4}$$

avec θ_{eq} la position d'équilibre, K la raideur, I le moment d'inertie et σ l'amortissement. Compte tenu du signe de l'accélération, la vitesse $\dot{\theta}$ devrait converger vers la valeur critique $\dot{\theta}_c(\theta)$ en même temps que le système converge vers l'attracteur. Nous avons tracé dans les figures 2.2 et 2.3 (a-c) la droite correspondant à $\dot{\theta}_c(\theta)$. Nous voyons que pour une raideur et un amortissement tel qu'il y a peu d'oscillations autour de l'attracteur, la vitesse $\dot{\theta}$ suit bien la droite $\dot{\theta}_c(\theta)$. Dans le profil dynamique moyenné (Figure 2.3 et 2.2 (a-c)), on peut observer des phases d'accélération sur les bordures de l'espace d'état. Il s'agit d'un effet du moyennage car, en pratique, chaque trajectoire présente une phase d'accélération localisée en θ_0, le point de départ de la trajectoire, et qui dépend de l'inertie. L'inertie implique aussi de possibles oscillations autour de l'attracteur (e.g. fig. 2.2a). Ces deux effets sont plus ou moins présents et marqués selon les paramètres du contrôleur (raideur K et amortissement σ). Le profil en vitesse obtenu est comparable aux profils obtenus en dérivant les champs de neurones dynamiques (cf. Section 1.3.1). Ainsi, pour obtenir le champs d'activation Per correspondant à un champ de neurones dynamiques, nous pouvons intégrer spatialement le profil dynamique obtenu. On cherche donc Per tel que $Ac = \nabla Per$. A un facteur et au signe près (que nous avons laissés de côté pour simplifier et comparer avec le champ d'activation des DNF), il s'agit bien du champ de perception Per décrit à la section 1.4.1. Dans le cas particulier qui nous intéresse, le champ $\mathbf{Per}$ est un vecteur dont les composantes sont liées à la discrétisation de l'espace d'état (2.5).

$$\forall k, Per_k = \int_{[0,k/n]} Ac(\theta)d\theta + cste \tag{2.5}$$

soit, suivant notre discrétisation de l'espace d'état :

$$\forall k, Per_k = \sum_{i=1}^{k} (Ac.Sen^\top)_i \cdot 1/n + cste \tag{2.6}$$

La constante d'intégration $cste$ est choisie de telle sorte que $\max_k(Per_k) = 1$. Le vecteur $\mathbf{Per}$ représente l'attracteur perceptif suivant la distribution de la vitesse en fonction de la position

θ (Fig. 2.2, 2.3 et 2.4 (d-f)). Les profils obtenus sont similaires aux profils des commandes en vitesse utilisés pour les champs de neurones dynamiques (voir Section 1.3.1). Le vecteur **Per** peut être interprété comme une fonction $Per(\theta)$ dépendant de θ. Le résultat de l'intégration de cette fonction est équivalent à l'activité d'un Champ de Neurones Dynamiques. Les paramètres du contrôle moteur permettent de moduler le profil obtenu. L'amplitude des activités du champ perceptif, donnant la force de l'attracteur, augmente avec la raideur K (Fig. 2.3) et diminue quand l'amortissement σ augmente (Fig. 2.2). Nous avons aussi étudié l'influence du ratio entre la raideur K et le coefficient d'amortissement (Fig. 2.4). A ratio constant, augmenter la raideur et l'amortissement permet d'augmenter l'amplitude des activités du champ dynamique équivalent et la stabilité de l'attracteur.

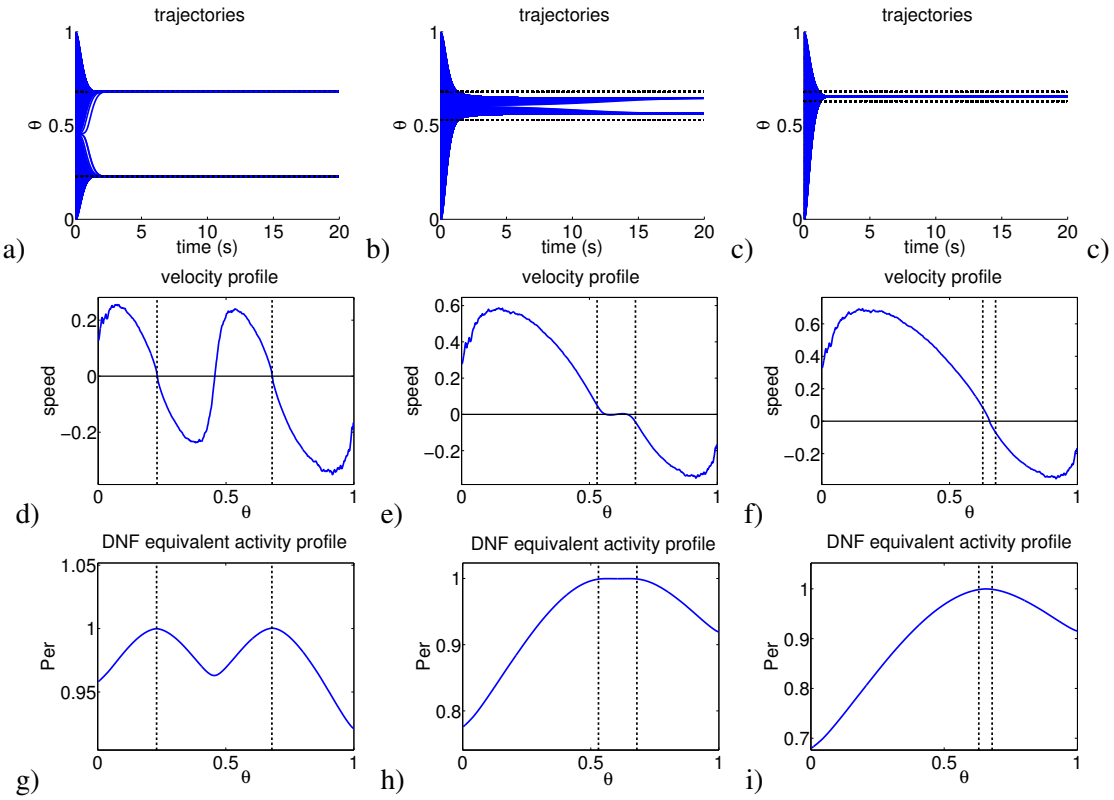


FIGURE 2.5 – Propriétés de bifurcation du contrôleur proposé. La rangée supérieure (a-c) montre les trajectoires (lignes bleues) et les deux positions attractrices appris (lignes pointillées noires). La rangée du centre (d-f) affiche le profil de vélocité angulaire en fonction de la proprioception θ. La rangée inférieure (g-i) donne le champ perceptif correspondant à l'intégration du profil de vélocité. De gauche à droite, les attracteurs appris sont de plus en plus proche pour aboutir à la fusion en un seul attracteur comportemental.

Nous avons ensuite utilisé l'outil mis en place pour étudier les propriétés de bifurcation du contrôleur proposé. Nous considérons uniquement le contrôle moteur de la proprioception (le discours peut être généralisé au cas où le contrôleur est visuomoteur avec les attracteurs activés à la fois par la vision et la proprioception). Chaque attracteur moteur est activé par l'état proprioceptif associé. Nous avons étudié le champ de perception obtenu dans le cas de deux attracteurs définis. Nous utilisons l'équation de catégorisation sensorielle (2), introduite dans

l'article de la Section 2.1, pour obtenir l'activation normalisée d'un état proprioceptif O_i^P. Dans le cas présent, l'équation se simplifie en l'Eq. (2.7).

$$O_i^P = \frac{S_i^P}{\sum S^P} \text{ with } S_i^P = e^{(-\frac{(\theta - W_i^P)^2}{2\beta^P})} \tag{2.7}$$

La reconnaissance d'un état proprioceptif S_i^P suit une mesure de similarité entre la proprioception actuelle θ (une seule dimension) et la proprioception apprise W_i^P. Cette mesure est basée sur une équation gaussienne de variance β^P. La sortie de la catégorisation proprioceptive O^P correspond à la normalisation des reconnaissances S^P. Dans le cas du contrôleur proprio-moteur, l'activation d'un attracteur moteur est directement liée à la reconnaissance de l'état proprioceptif associé.

Notre objectif est d'observer la bifurcation entre les deux attracteurs sur le champ perceptif. Suivant le paramètre de covariance β^P, les états proprioceptifs répondent plus ou moins fortement pour une même position. Avec un paramètre de covariance faible, les deux attracteurs appris sont suffisamment éloignés l'un de l'autre pour que seul un attracteur réponde. Le comportement qui apparaît correspond bien aux deux attracteurs encodés (Fig. 2.5 (a,d,g)). Si les attracteurs sont plus proches, ou si le paramètre de covariance β^P de la gaussienne (2.7) est plus grand, alors les attracteurs appris sont fusionnés (Fig. 2.5 (c,f,i)). Le comportement qui en découle correspond à un seul attracteur situé entre les attracteurs appris. La propriété de bifurcation des DNF est donc bien présente dans le modèle que nous proposons.

2.3 Bilan du chapitre

Dans ce chapitre, nous avons proposé le modèle Dynamic Muscle PerAc (DM-PerAc) permettant de construire un contrôleur visuomoteur neuronal s'appuyant sur un codage exploitant des activations musculaires. Le contrôle est fait articulation par articulation en considérant les activations de muscles agonistes et antagonistes. Selon les activations musculaires, le contrôleur peut générer des bassins d'attraction sur des trajectoires ou sur des points fixes. Pour ce dernier cas, nous avons proposé un apprentissage basé sur l'adaptation des coefficients d'activation des muscles. Des catégories sensorimotrices recrutées fixent les contraintes que doivent respecter les mouvements lorsque l'attracteur associé est actif. En fonction des mouvements produits, les coefficients d'activation sont augmentés ou diminués, contraignant progressivement le bassin d'attraction pour qu'il corresponde à la posture désirée. La construction du bassin d'attraction est dynamique et progressive au fur et à mesure des actions (et des erreurs) faites par le robot.

En navigation, l'information sur la position 2D que va rejoindre le robot mobile ne peut pas être connue à partir d'un seul couple lieu/action générant une orientation. C'est la contribution des différents couples qui va permettre de construire un bassin d'attraction vers un endroit donné. De la même manière, l'information sur la posture que va prendre le bras robotique ne peut pas être connue à partir de l'activation d'un seul muscle. C'est la combinaison des activations des différents muscles qui permet de construire un bassin d'attraction sur une posture particulière. C'est particulièrement important si l'on considère les actuateurs qui peuvent permettre de simuler artificiellement des muscles comme par exemple les actuateurs hydrauliques ou pneumatiques. En effet, les commandes motrices bas niveau peuvent déterminer les pressions appliquées dans les vérins donnant alors les mouvements des articulations. Ainsi, la position et

la dynamique du mouvement résultent autant de la mécanique et de la disposition des actuateurs que du contrôleur. Le modèle que nous proposons peut permettre l'apprentissage du contrôle quelque soit le type d'actuateur utilisé. L'apprentissage s'appuie sur le comportement du robot, c'est à dire que c'est bien par ses actions que le robot peut apprendre comment adapter son comportement. En navigation, la contrainte guidant l'apprentissage doit être donnée par un humain, le robot apprend à ne pas s'éloigner d'une trajectoire ou d'une position donnée. Pour le contrôle d'un bras, nous proposons que cette contrainte soit donnée par la catégorisation de la configuration proprioceptive désirée.

Les auteurs de [Ganesh and Albu-Sch, 2010] ont proposé un modèle de contrôle qui peut adapter en ligne les activations musculaires (paramètres du contrôle) afin de compenser des perturbations externes. Dans notre modèle DM-PerAc, la modification des activations musculaires a été mémorisée pour apprendre le contrôle visuomoteur. Cet apprentissage a lieu juste après le recrutement de la nouvelle catégorie proprioceptive donnant la contrainte pour l'apprentissage. Deux améliorations seraient donc possibles : le système devrait pouvoir adapter temporairement le contrôle sans apprentissage, i.e. sans modification des poids synaptiques dans le réseau, et il devrait être possible de reprendre l'apprentissage d'un attracteur à n'importe quel moment. Grâce à l'adaptation, les perturbations externes pourraient être compensées sans autant modifier ce qui a été appris. Selon les circonstances, cette adaptation pourrait être à l'origine d'un apprentissage. En particulier, on pourrait envisager l'utilisation de contextes pour apprendre des commandes motrices différentes selon la masse des objets manipulés.
Il devrait aussi être possible de modifier les attracteurs construits même après la première phase d'apprentissage. Ceci impliquerait de mettre en place des processus qui évalueraient la nécessité et la façon de changer les poids des associations sensorimotrices (e.g. nouvel attracteur avec d'autres dépendance sensorielles ou pas). Ainsi, dans le modèle DM-PerAc (plus précisément Eq. (20) dans l'article Section 2.1), l'occurrence d'un mouvement erroné avec adaptation de l'encodage et la prédiction de cette occurrence ont permis de déterminer si l'apprentissage devait être poursuivi ou non. Cette solution relativement frustre pourrait être améliorée en considérant la notion de progrès comme dans les modèles de curiosité [Schmidhuber, 1991; Oudeyer and Kaplan, 2004]. Notamment, l'un des principes clés de la curiosité artificielle est que le robot détermine ce qui doit être appris avec la possibilité de *reprendre et poursuivre l'apprentissage d'une compétence*.

Dans la Section 2.1, nous avons proposé d'utiliser la méthode "Yuragi" [Fukuyori et al., 2008] pour permettre l'exploration et l'apprentissage d'attracteurs sensorimoteurs pertinents. Cette méthode peut être utilisée pour restreindre le nombre total d'attracteurs recrutés tout en assurant une bonne précision et réactivité dans les conditions qui le nécessitent. D'autre part, nous avons proposé d'étendre cette propriété pour apprendre des trajectoires dynamiques via la combinaison d'activations musculaires adéquates. Nous pouvons aussi considérer la possibilité de réutiliser les attracteurs préexistants pour encoder à un niveau supérieur des "macro-attracteurs" s'appuyant sur les attracteurs de plus bas niveau. Un macro-attracteur activerait de manière adéquate plusieurs attracteurs pour obtenir le comportement moteur désiré. Étudions l'utilisation de la méthode "Yuragi" pour réaliser cet apprentissage sur les attracteurs. Le terme stochastique dans l'équation de Langevin (Eq. (22) dans l'article de la Section 2.1) permettrait explorer l'espace d'état et de rejoindre la position que l'on souhaiterait apprendre. Dans le cas idéal, le modèle inverse appris permettrait d'interpoler correctement la combinaison motrice à partir de la reconnaissance des catégories visuelles ou proprioceptives associées aux attracteurs.

Ainsi, l'apprentissage des activations des catégories sensorielles quand le système est à la position désirée suffirait à apprendre le macro-attracteur. Cependant, en pratique, l'interpolation à partir du modèle inverse appris sur un certain nombre de positions ne sera pas parfaite. Pour obtenir la bonne combinaison d'attracteurs, il faudra adapter les activations de ces différents attracteurs directement. Nous faisons l'hypothèse qu'un mécanisme similaire à celui apprenant les combinaisons d'activations musculaires dans l'article (Section 2.1) pourrait résoudre l'apprentissage des "macro-attracteurs".

Dans la section 2.2, nous avons proposé une méthode permettant de visualiser les mouvements générés par le contrôleur DM-PerAc sous la forme d'un champ d'activation dit "champ perceptif" équivalent à un champ de neurones dynamiques (DNF, [Schoner, 1995]). Nous avons pu constater la similarité entre les profils d'activation ainsi que le fait que le contrôleur DM-PerAc vérifie aussi la propriété de bifurcation au niveau des attracteurs construits. L'avantage du contrôleur DM-PerAc est d'éviter le codage en population, permettant un gain en terme de temps de calcul et un apprentissage des propriétés spécifiques des attracteurs. Dans [Andry et al., 2004], le DNF est dérivé pour obtenir le profil de commande en vitesse du mouvement d'un bras. Une lecture du profil de vitesse à partir de la valeur de l'état courant du système donne la commande en vitesse. Dans notre modèle, nous évitons donc à la fois la construction du champ neuronal complet et la dérivation de ce champ pour obtenir la commande motrice. D'autre part, il est plus facile de modifier le profil des attracteurs dans le cas du modèle DM-PerAc que dans le cas DNF. En effet, en utilisant un champ de neurones avec des bulles d'activité gaussienne, un compromis est nécessaire entre la largeur du bassin d'attraction et la dynamique au niveau de l'attracteur. Dans le cas d'un bassin large, la dynamique est alors très faible autour de l'attracteur et donc l'attracteur est peu robuste. Dans l'architecture que nous proposons, le champ d'attraction n'est borné que par les limites de l'espace d'état. De plus, la dynamique peut être modulée simplement en jouant sur la raideur et l'amortissement. Ces deux paramètres devraient pouvoir être modifiés en ligne pour ainsi adapter la stabilité de l'attracteur en fonction des situations. Le contrôleur DM-PerAc ne reprend pas toutes les propriétés des DNF. Des mécanismes pourraient être inclus pour compléter les propriétés du contrôleur DM-PerAc. Par exemple, considérons le fait d'avoir un champ d'attraction limité dans l'espace. Une solution pour obtenir cette propriété serait d'avoir un seuil sur la reconnaissance des états proprioceptifs en dessous duquel l'attracteur correspondant ne serait pas activé. La propriété de mémoire n'est pas non présente dans le modèle DM-PerAc. Pour pouvoir l'introduire, il faut envisager d'inclure une modulation des états proprioceptifs dépendant des activités temporelles passées et ainsi introduire une hystérésis. Nous considérons ainsi qu'une partie de ces propriétés seraient plus liées à des mécanismes attentionnels [Johnson et al., 2009; Fix et al., 2010] qu'à des mécanismes essentiellement moteurs.

Dans le processus de visualisation du champ perceptif, le rôle de l'inertie sur les mouvements est partiellement occulté par l'effet du moyennage entre toutes les trajectoires. En pratique, cependant, le mouvement obtenu dépend de l'inertie du système avec une phase d'accélération et de décélération voire des oscillations autour de l'attracteur. Dans le cas des champs de neurones dynamiques, le profil en vitesse présente une phase d'accélération et de décélération ; la dynamique d'un mouvement complet est donnée par le profil. Pour autant, l'approche du contrôle moteur consistant à utiliser le profil dynamique dérivé des champs de neurones dynamiques comme commande motrice est limitée. Outre la difficulté d'adapter la dynamique du contrôle selon le point de départ du mouvement, cette dynamique occulte aussi l'effet des propriétés

physiques et mécaniques de l'actuateur (e.g. son inertie réelle). En effet, il nous semble que c'est la prise en compte de ces propriétés dans l'apprentissage du contrôle moteur qui permettra d'améliorer la robustesse du contrôleur. Dans notre méthode de comparaison, nous avons utilisé la vitesse comme critère pour représenter l'action Ac et ainsi visualiser le champ perceptif. L'avantage de ce critère est de suivre le cadre des comportements dynamiques défini dans [Schoner, 1995]. Cela permet aussi d'obtenir un champ perceptif comparable avec les champs de neurones dynamiques.

De précédents travaux [Andry et al., 2004] ont montré que l'homéostasie associée à l'ambiguïté de la perception suffisait pour permettre l'émergence d'un comportement d'imitation de très bas niveau. Le contrôleur visuomoteur DM-PerAc vérifie ces propriétés et peut donc permettre l'émergence d'un comportement social. Nous nous intéresserons dans le chapitre suivant à la possibilité d'obtenir des capacités d'apprentissage en interaction de plus en plus sophistiquées. Le robot pourra aussi s'engager dans des comportements sociaux. Nous verrons comment ces compétences peuvent être obtenues en considérant les associations sensorimotrices construites par différents niveaux de modèles développés pour la navigation (transitions entre états, séquences, carte cognitive et planification).

Communications dans des ateliers de travail internationaux

de Rengervé, A., Andry, P., and Gaussier, P. (2011). A Neural Network Generating Force Command for Motor Control of a Robotic Arm. In *International Workshop on bio-inspired robots*, Nantes, France

Les choses de l'esprit qui ne sont pas passées par les sens sont vaines.

– Leonard de Vinci

CHAPITRE 3

De l'imitation immédiate vers la coopération Homme-robot

Nous avons présenté, au chapitre 1, l'approche PerAc qui permet de construire des bassins d'attraction sensorimoteurs en associant des cellules de lieux et des orientations de mouvement pour un robot mobile. Ce modèle permet à un robot de suivre une trajectoire ou de rejoindre une position particulière de l'espace. Dans le chapitre 2, nous avons montré que l'approche PerAc permettait de construire un contrôleur visuomoteur pour un bras robotique en suivant les mêmes mécanismes que pour la navigation. Nous allons maintenant étudier l'extension du modèle pour l'apprentissage de tâches en interaction avec un professeur humain. D'autre part, le modèle devra permettre d'apprendre des taches complexes qui nécessiteront que le comportement dépende de buts et de contextes motivationnels. Les différentes capacités acquises seront mises en oeuvre dans des comportements sociaux et nous aborderons la question de la coopération Homme-robot.

Dans la section 3.1, nous présenterons le modèle DM-PerAc pour l'apprentissage de tâches avec un professeur humain. Le but sera d'obtenir un robot capable d'imiter en simultané (Sec. 3.1.2) et en différé (Sec. 3.1.1) des gestes observés, et qu'il puisse apprendre des séquences de gestes lors d'une démonstration par manipulation passive de son bras. Le robot pourra planifier la séquence à réaliser en fonction d'un contexte basé sur les informations extraites de l'environnement (Sec. 3.1.3). Le modèle complet s'appuiera sur une modélisation de différentes structures cérébrales initialement utilisées en navigation (Sec. 3.1.4, article 2). Les mécanismes d'apprentissage sont non spécifiques à la tâche de navigation. Ils permettront de contrôler un bras robotique pour réaliser les différents comportements d'apprentissage de tâches et d'imitation. La tâche robotique principale sera l'apprentissage en ligne d'une séquence de gestes pour attraper un objet (e.g. canette) et le déplacer à un autre endroit, en fonction de sa couleur par exemple.

Suivant le principe de la référenciation sociale [Feinman, 1982; Thomaz et al., 2005; Boucenna et al., 2010b], un enfant peut s'appuyer sur des indices sociaux comme l'expression faciale de ses parents (e.g. sourire (positif) ou expression de peur (négatif)) pour évaluer l'intérêt ou le danger d'un objet ou d'une situation. Cette évaluation peut aussi s'appliquer à l'action réalisée, par exemple l'orientation du mouvement en navigation [Hasson et al., 2010]. Cette approche sera utilisée pour guider l'apprentissage des "bonnes" séquences pour trier des objets. Dans la section 3.1, nous utiliserons uniquement des retours sociaux positifs équivalents à donner des récompenses pour une séquence correctement réalisée. Puis, dans la section 3.2 et l'article 3, nous nous intéresserons à la possibilité d'utiliser des renforcements négatifs pour adapter le com-

portement en inhibant les buts incorrects. La difficulté portera sur la mise en place du processus permettant au système de déterminer le but caractérisant le comportement. Le modèle proposé sera validé en simulation dans une expérience de navigation motivée (Section 3.2.1) puis il sera utilisé pour modifier le comportement du robot dans une tâche de tri de canettes (Section 3.2.2).

3.1 Apprentissages sensorimoteurs et capacités de planification pour l'apprentissage de tâche et l'imitation

3.1.1 Imitation immédiate de gestes

Dans [Gaussier et al., 1998; Andry et al., 2004], les auteurs ont montré qu'un comportement d'imitation immédiate pouvait émerger de la combinaison d'un homéostat visuomoteur appris et de l'ambiguïté de la perception. Ainsi, un bras robotique peut imiter en simultané les gestes d'un humain présent devant lui. Nous verrons dans la section III-B de l'article 2 (Section 3.1.4) que suivant les mécanismes décrits ci-dessus, le contrôleur visuomoteur proposé au chapitre 2 permettra d'obtenir un comportement d'imitation immédiate.

Dans une première phase, le système apprendra les associations entre les informations visuelles et motrices pour construire l'homéostat visuomoteur. A cause de ses capacités visuelles limitées, le robot ne sera pas capable de discriminer sa propre main de la main de l'humain (ambiguïté de la perception). En conséquence, le robot pourra percevoir la main de l'humain et la considérer comme sa propre main. Plusieurs travaux en psychologie [Nielsen, 1963; Fourneret and Jeannerod, 1998; Jeannerod, 1999] ont montré des comportements humains similaires. Dans les expériences, le sujet devait suivre avec un stylo une ligne tracée. Il ne voyait sa main qu'à travers un système de miroirs transparents ou non. En cours d'expérience, le miroir pouvait refléter la main d'un autre individu. Le sujet pouvait voir la main de l'autre individu mais continuait de considérer qu'il s'agissait de la sienne. Il corrigeait alors ses mouvements en fonction des erreurs observées, produites par l'autre individu, plutôt qu'en s'appuyant sur le retour proprioceptif de sa propre main. Dans notre système robotique, l'architecture de contrôle implémentera un homéostat visuomoteur, donc le système tendra à maintenir un équilibre entre les informations visuelles et proprioceptives. Si une différence est perçue, alors le système agira pour revenir à l'état d'équilibre : le robot bougera son bras de telle sorte que la configuration proprioceptive corresponde au stimuli visuel perçu en accord avec l'apprentissage sensorimoteur préalable. Ces mouvements de corrections produiront l'imitation des gestes du démonstrateur [Andry et al., 2004; de Rengervé et al., 2010a]. Grâce à cette approche de l'imitation, le problème de correspondance [Nehaniv and Dautenhahn, 2002] sera évité car le robot imitera ce qui est observé selon ses propres capacités.

3.1.2 Imitation en différé et apprentissage par observation

Notre contrôleur visuomoteur DM-PerAc et des capacités d'apprentissage de séquences basées sur un modèle utilisé en navigation (Section 1.5.1) permettront à notre robot d'imiter en différé. Le robot pourra ainsi réaliser des comportements d'apprentissage par observation.

Le mécanisme de mémorisation des séquences est basé sur un modèle de l'hippocampe, lui-même inspiré d'un modèle du cervelet. Les travaux initiaux sont ceux de [Banquet et al., 1997] pour l'apprentissage de séquences de cellules de lieu et de transitions entre ces cellules

de lieu [Gaussier et al., 2002]. Dans le cadre d'un contrôle de bras robotique, [Lagarde et al., 2010] ont utilisé des catégories motrices comme entrée d'un modèle similaire pour apprendre en passif des séquences temporelles de gestes reproduites en différé. Nous avons utilisé une version plus récente du modèle de prédiction d'événements et de transitions proposée par [Hirel et al., 2010]. L'architecture pour l'apprentissage des transitions s'appuie sur le cortex entorhinal (EC), le gyrus dentelé (DG) et les champ 1 et 3 (CA1/CA3) de la corne d'Ammon (figure 3 de l'article 3, Section 3.2.1). Des catégories sensorielles sont apprises au niveau cortical. Le cortex entorhinal fait l'interface entre le cortex et l'hippocampe. Son rôle est aussi de détecter les changements d'états. Le gyrus dentelé présente une structure neuronale granulaire. Les activations neuronales des groupes de neurones permettent de définir une base temporelle. L'aire DG fournit donc une mémoire temporelle pour les apprentissages d'évènements. L'aire CA3 apprend à prédire les changements d'état à partir de l'information contenue dans DG. Le signal temporel produit par CA3 suffit pour la reproduction de séquences simples. Nous laisserons le lecteur se reporter à [Hirel, 2011; Hirel et al., 2013] pour les équations détaillées de l'apprentissage temporel entre EC, DG et CA3. Dans cette thèse, nous nous intéresserons principalement à l'aspect spatial de l'apprentissage à savoir la connaissance de l'état futur pour reproduire une séquence donnée. Lors des expériences, l'information temporelle servira uniquement à limiter le temps accordé pour rejoindre un état particulier. Si l'état n'est pas rejoint dans le temps prévu, un rebouclage de la prédiction de l'état suivant simule un passage réussi afin d'enchaîner la séquence sans rester indéfiniment en attente. Dans les expériences présentées, les mouvements réalisés par le robot seront plus rapides que ceux montrés et donc cette limite temporelle n'interviendra que lorsque la configuration prédite ne peut être atteinte par le robot.

En utilisant les informations visuelles en entrée, le système peut prédire la catégorie visuelle suivante. Grâce au contrôleur visuomoteur construit au chapitre 2, le robot produira les mouvements pour rejoindre les positions visuelles prédites. Afin que le robot puisse apprendre par observation, le système doit pouvoir *inhiber ses propres mouvements durant la démonstration*. Le robot, immobile, observera une séquence de gestes et apprendra la succession des états visuellement activés. Puis, lorsqu'il rejouera la séquence visuelle apprise, le contrôleur visuomoteur permettra de générer les commandes motrices reproduisant la séquence observée suivant les capacités du robot. On montrera donc que *la combinaison de la capacité à inhiber ses actions et à mémoriser des séquences visuelles avec le contrôle visuomoteur développé précédemment suffisent à obtenir un comportement d'imitation différée i.e. d'apprentissage par observation*. La section IV de l'article 2 (Section 3.1.4) présentera l'expérience réalisée pour valider ce principe. Le robot apprendra par observation une tâche dans laquelle il doit réaliser la séquence de gestes permettant d'atteindre une canette, de l'attraper (grâce à un réflexe de préhension) et de la déplacer.

3.1.3 Apprentissage de séquences de gestes motivées avec capacités de planification

Notre modèle doit maintenant être complété afin qu'il puisse apprendre des buts associés à des contextes motivationnels, qui lui permettront de planifier ses comportements. Nous utiliserons un apprentissage de séquences incluant des associations avec des contextes motivationnels selon le principe de construction des cartes cognitives utilisées en navigation (Sec. 1.5.1). Les séquences seront donc encodées dans un graphe de transitions entre états. Dans le modèle

de [Hirel, 2011], la région CA3 possède des neurones prédicteurs d'événements futurs qui ne différencient pas les transitions. Les transitions sont codées dans CA1. Les neurones de CA1 possèdent des connexions provenant de CA3 et de EC. On peut donc y fusionner les informations de prédiction provenant de CA3 avec des informations sur le dernier événement provenant de EC (on suppose l'existence d'une mémoire identifiant le dernier événement dans EC, qui projette vers CA1). Le recrutement dans CA1 est alors déclenché par la neuromodulation ACh (Acétylcholine) correspondant à l'arrivée d'un nouvel événement. Les neurones recrutés prédisent les transitions possibles depuis un état donné. L'apprentissage, déclenché pour le neurone recruté ou le neurone ayant la plus forte activité si celle-ci est au dessus du seuil de recrutement, est régi par l'équation suivante :

$$\frac{dW_{ij}(t)}{dt} = f(ACh(t) \cdot (\alpha \cdot X_j(t) - \gamma \cdot W_{ij}(t))) \tag{3.1}$$

avec f une fonction rampe telle que $f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } 0 \leq x < 1 \\ 1 & \text{si } x \geq 1 \end{cases}$

$ACh(t)$ est la neuromodulation d'apprentissage (1 si un événement a lieu, 0 sinon) basée sur l'acétylcholine, α la vitesse d'apprentissage et γ un facteur d'oubli. L'équation pour le calcul de l'activité dans CA1 est :

$$X_i^{CA1}(t) = f(\sum_j W_{ij} \cdot X_j - \theta) \tag{3.2}$$

θ est le seuil d'activation utilisé pour inhiber les neurones non co-activés par EC et CA3. Les X_j et W_{ij} représentent indifféremment les activités et les connexions synaptiques provenant de EC et CA3. Lors d'une transition entre 2 lieux, seuls les neurones co-activés correspondant au code de cette transition déchargent. Cela permet de mettre en place un mécanisme de recrutement dans CA1 se basant sur un seuil d'activité minimale, identifiant le fait qu'un neurone codant pour la transition a déjà été recruté. Enfin, grâce à la modulation, la transition effectuée peut être propagée vers la carte cognitive pour être liée aux autres transitions. D'autre part, les transitions prédites T^P sont transmises à la partie sélectionnant les actions devant être réalisées.

Nous utiliserons les mécanismes des cartes cognitives pour apprendre des séquences de transitions et leur lien avec la satisfaction d'un but donné. Contrairement à l'usage classique d'une carte cognitive, il sera important que le robot n'apprenne que des chemins spécifiques de manière à ne pas créer de raccourcis inopportuns lors de la reproduction des séquences apprises. Les avantages qui nous intéresseront sont la possibilité de réutiliser des séquences déjà apprises présentes dans le graphe, la possibilité de modifier le comportement du robot via l'adaptation des associations entre les contextes motivationnels, les buts et le graphe de transitions.

Nous supposerons qu'il existe au niveau préfrontal une mémoire de travail des deux dernières transitions. Les connexions récurrentes sur la carte cognitive sont renforcées entre les neurones O_i et O_j codant pour ces transitions (3.3). Lors de l'exploration de l'environnement, la carte cognitive est donc progressivement créée au fur et à mesure que le robot passe d'un état à un autre. L'équation régissant l'apprentissage du graphe est la suivante :

$$\frac{dW_{ij}^{rec}(t)}{dt} = T(t) \cdot ((\gamma - W_{ij}^{rec}(t)) \cdot O_i(t) \cdot O_j(t) - W_{ij}^{rec}(t) \cdot (\lambda_1 \cdot O_j(t) - \lambda_2)) \tag{3.3}$$

avec $T(t)$ un signal binaire (0 ou 1) actif lorsqu'une transition est effectuée (passage d'un

état à un autre). Ce signal contrôle l'apprentissage des connexions récurrentes W^{rec}. γ est un paramètre appartenant à $[0, 1]$ et réglant la diffusion des activités dans la carte. λ_1 et λ_2 sont des paramètres d'oubli actif et passif, respectivement, sur les connexions récurrentes.

Les neurones de la carte cognitive peuvent être associés avec des contextes motivationnels $M_k(t)$ ou des buts. Ces contextes participent donc à l'activité de sortie O des neurones de la carte cognitive via l'activité motivationnelle O^{motiv}. Nous considérerons deux versions pour l'apprentissage et le calcul de cette activité O^{motiv}.

Cas 1 : La première version, présentée dans l'article 2 section 3.1.4, s'appuie sur une association directe avec les contextes motivationnels (W_{ik}^{motiv}, équation (9) dans la Section V-A de l'article 2). L'équation d'apprentissage est reprise ci-dessous :

$$\Delta W_{ik}^{motiv}(t) = R(t) \cdot M_k(t) \cdot (\alpha^m \cdot (1 - W_{ik}^{motiv}(t)) \cdot O_i(t) - \gamma^m) \tag{3.4}$$

avec α^m un facteur d'apprentissage, λ^m un facteur d'oubli et R(t) un signal marquant la réception d'une récompense. L'activité motivationnelle O^{motiv} est obtenue par :

$$O_i^{motiv}(t) = f(\max_k(W_{ik}^{motiv}(t) \cdot M_k(t)) \tag{3.5}$$

Ces deux équations (3.4) et (3.5) permettent d'apprendre des buts quand une récompense est reçue en renforçant l'association entre le contexte motivationnel actif et la dernière transition réalisée. Les précédentes associations pour ce contexte motivationnel sont progressivement oubliées.

Cas 2 : Dans la deuxième version de l'apprentissage des buts, une couche neuronale intermédiaire, entre les contextes motivationnels et la carte cognitive, représentera les transitions-buts apprises. Les potentiels d'activité transiteront donc depuis les contextes motivationnels vers la carte cognitive en passant par cette couche intermédiaire. Une modulation des neurones de cette couche permettra de détecter le but poursuivi et ainsi d'exploiter les signaux de renforcement négatifs pour le désapprendre son association en tant que but (sec. 3.2). Les équations pour cet apprentissage des buts et leurs associations avec la carte seront données dans l'article 3 (sec. 3.2.1).

Dans les deux cas, le traitement réalisé permettra d'obtenir l'activité O^{motiv} liée au contexte motivationnel qui sera utilisée pour calculer la sortie O des neurones de la carte cognitive.

$$\begin{aligned} O_i(t) = \quad & (1 - R(t)) \cdot f(\max_j(W_{ij}^{rec}(t) \cdot O_j(t), O^{motiv}(t))) \\ & + R(t) \cdot [X_i^{mem1}(t) + T(t) \cdot X_i^{mem2}(t)] \end{aligned} \tag{3.6}$$

X^{mem1} est l'activité provenant de la mémoire stockant la dernière transition effectuée, tandis que X^{mem2} fournit l'information de l'avant dernière transition effectuée. $R(t)$ est un signal marquant la réception d'une récompense. L'activité O des neurones de la carte cognitive correspond soit à la propagation des activités des contextes motivationnels en phase d'utilisation, soit exclusivement aux activités de la ou des deux dernières transitions réalisées lors d'un apprentissage (lorsqu'une transition est réalisée ou une récompense est obtenue). La carte cognitive construite contiendra les gradients propagés par les buts actifs. Le potentiel des neurones dans la carte viendra ensuite biaiser la sélection de la transition à réaliser (compétition) au niveau des ganglions de la base. Les transitions seront associées avec l'état final du changement d'état. Ainsi, les catégories sensorielles (proprioceptive, et donc visuomotrice) correspondant au mouvement planifié seront sélectionnées permettant la génération des commandes motrices adéquates.

Pour que l'apprentissage puisse s'effectuer dans les cas où la tâche n'implique pas une satisfaction directe de besoins élémentaires, le signal de récompense sera associé à un contexte social par exemple la reconnaissance d'un sourire. On se trouvera ainsi dans un contexte proche d'une référenciation sociale [Feinman, 1982]. En pratique, dans les expériences, nous utiliserons une reconnaissance simplifiée (basée sur la couleur d'une image de visage souriant ou fâché) pour extraire le type de retour (positif ou négatif). Néanmoins, les travaux de [Boucenna et al., 2010a; Boucenna, 2011] ont montré qu'un robot pouvait apprendre à reconnaître des expressions faciales via la reconnaissance des états internes associés à ces expressions. Ainsi, un visage souriant pouvait être reconnu comme positif et un visage triste ou inquiet comme négatif.

Les premiers résultats, qui utilisent juste des retours sociaux positifs, seront donnés dans la section V de l'article 2 (Section 3.1.4). L'expérience réalisée sera l'apprentissage en ligne et rapide, avec un professeur humain, d'une tâche de tri de canettes en fonction de leur couleur. Durant l'apprentissage, le professeur humain enseignera les séquences de gestes en manipulant le bras robotique en passif et fournira les récompenses validant le succès d'une séquence.

3.1.4 Article2 : une architecture neuronale inspirée des structures du cerveau pour l'apprentissage de tâches et l'imitation

de Rengervé, A., Andry, P., and Gaussier, P. (2013a). Autonomous Development of Gesture and Action Imitation as a Result of Sensorimotor Learning and Planning in a Brain Based Architecture

Autonomous Development of Gesture and Action Imitation as a Result of Sensorimotor Learning and Planning in a Simple Neural Architecture.

Antoine de Rengervé, Pierre Andry, and Philippe Gaussier

***Abstract*—This paper proposes to view human-robot cooperation as the result of the development of a Neural-Network control architecture. The general principles of the model come from previous experiments in navigation and underline the importance of the coding of the associations describing the behaviors. We propose that behaviors can be considered as sensorimotor attractors, and that they can be learned by a control architecture under the form of sets of multimodal perception-action associations. From this starting point, we show how the ambiguity of the perception allows the emergence of low-level imitative behavior, which can be viewed as an explicative model for higher level behaviors such as deferred imitation, and observational learning. Finally we show that our model can also be the basis of early human-robot cooperation and we illustrate this point with the learning and planning of tasks consisting in sorting different cans into separate boxes.**

***Index Terms*—multimodal integration through development; biology-inspired architecture for development; neural networks for development; development of social skills; robots with development and learning skills; using robots to study development and learning; coordination and integration of behaviors through development.**

I. INTRODUCTION

FOR more than 10 years, we have put our efforts in understanding imitation in social systems. We have build Artificial Neural Network (NN) control architectures in order to tackle the communication and learning function of imitation. Our work, following a developmental approach, along with collaboration with developmental psychologists, cognitive psychologists and neurobiologists have led us to understand that tasks and behaviors cannot be entirely defined, and are thus difficult to be directly coded, by the parameters of a control architecture. A behavior is built upon a wide range of interactions of different levels. A system learning a behavior has to capture the dynamical sensorimotor attractors describing this behavior, whose parameters are -unfortunately partly- linked to the control architectures, but also to the agent's embodiment, and to the interactions of this brain-body with the surrounding environment. In such conditions, the issues of how to learn, adapt and share these attractors are fundamental for one who wants to achieve natural and intuitive non verbal human-robot cooperation.

In this paper, we show that higher form of non-verbal human-robot cooperation such as task-learning by demonstration can be acquired through an epigenetic approach starting from self exploration and low-level imitation. In an abridged developmental procedure, we propose a NN architecture that links self exploration, low-level immediate imitation, deferred imitation, task learning by demonstration and higher level planning solution. The main property of this model is the ability to learn and follow behavioral attractors throughout non-verbal human-robot interactions. Interestingly, the whole model itself was not built to produce imitative behaviors, since it comes from the study of rodents abilities to navigate and application to an autonomous mobile robot. In the developmental process of our arm-eye robot, imitation is at the same time a side-effect of the architecture's coding, and the trigger of higher level behaviors. Reversing classical engineering approaches, we advocate that it is because there is at first no distinction from others that our system imitates. With an initial learning phase based on low-level associations between motor actions and sensations of self, perception ambiguity allows to play with visual information about others. A first low-level imitative behavior emerges and allows spontaneous human-robot interaction. Next, we show that motor inhibition and sequence learning allow to use the same model to explain observational learning, that is to say deferred imitation, and learn actions as sequences of moves demonstrated by others. Finally, we use the same associative mechanisms and draw a second parallel with a NN planning model used in navigation to select between different actions according to the satisfaction of different motivations.

Information coding is crucial, and whereas our associative learning rules are sub-optimal when compared to more classical reinforcement or stochastic optimization techniques, they tend to provide equal performances in the frame of population coding (such as the brain does) with the supplementary advantage of allowing information availability and reuse for other behaviors. This last point is crucial in our approach since it allows us to converge in navigation tasks, learning tasks and human-robot social interactions with the same model and the same general scientific issues.

II. PLURI-DISCIPLINARY MOTIVATIONS

The process of learning a new activity via interaction is long and complex. When engaged in such learning process, social species such as great apes and humans combine multiple strategies, whether the subject tries to solve the task by itself from previous observation, or whether he/she is involved in a more cooperative interaction with a caregiver or a demonstrator [1]. For example, authors in animal psychology have shown

ETIS UMR CNRS 8051, ENSEA, University Cergy Pontoise F-95000 Cergy Pontoise, France

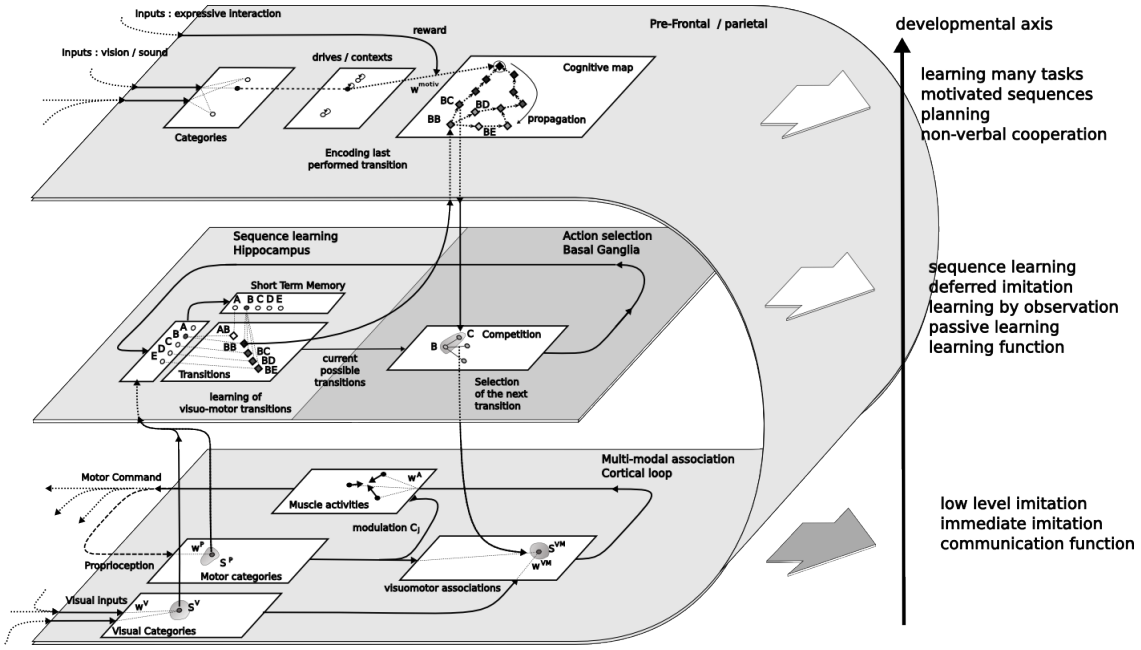


Fig. 1. Overview of the model. The developmental process lies on the progressive maturation and involvement of different cortical and subcortical loops in the brain. This architecture suggests how simple principles corresponding to the effects of these loops can be combined to explain different imitative behaviors from simple low level immediate imitation to reproduction of goal dependent sequences of gestures. The cortical loop comes across the parietal to premotor to motor cortices and learn sensorimotor associations. The spatio-temporal properties of the Hippocampal structures (Entorhinal Cortex, Dendate Gyrus, Cornu Ammonis (CA1, CA3)) can explain sequence learning capabilities. Learned transitions between states built at the cortical level can be selected to reproduce particular sequences. The action selection is performed in the Basal Ganglia (BG). Still, the role of BG is limited in our model (e.g. Reinforcement Learning should be included). The hippocampo-fronto-basal loop introduces goals built in the frontal cortex to bias the choice of the sequence to be performed.

that monkeys are able to take advantage from : exploration, stimulus enhancement [2], response facilitation [3] linking priming and affordances to emulation[4] and simple forms of imitation [5]. Among humans, and especially during the first 4 years of development, we know that the aforementioned mechanisms are exploited in conjunction with communicative functions [6] : contingencies detection and turn taking [7], alternating trials and errors through role switching. But it is not clear how these mechanisms are sufficient to explain imitative behaviors such as spontaneous imitation (imitation of gestures for communication purpose [8]) deferred imitation (also called learning by observation [9], that is to say the ability to reproduce a given set of actions to fulfill a goal from the previous observation of a demonstrator [10]), learning by demonstration, or cooperation [11]. If "imitation" is known to be a central marker of epigenesis, it remains at the same time a confusing umbrella term [12]. The multiplication of mechanisms, behaviors, levels, in conjunction with the different ways to learn the same task (by observation, by demonstration, etc.) have always made difficult the definition of a phenomenon that is non unitary. The difficulty is even greater if we think that social species are able to combine these behaviors in the same learning sequence.

Interestingly, if we want to build non-verbal robots able to adapt in dyadic and triadic situations, we face the necessity to build such a versatile architecture. The same controller must be able to exhibit the appropriate behavior during a cooperation process. From our experience in epigenetic robotics, such architecture implies to take into account motor control [13], [14], multimodal fusion and sensorimotor coordination (especially vision and proprioception) [15], [16], [17], temporal sequence learning [18], [19], [20], motivations, reward and contextual information. This raises the question of the existence of a limited range of mechanisms or principles that would allow to get for free, with the same model, all the learning and communication power of imitative behaviors. We propose that such model can be built from simple and multiple associative mechanisms, allowing to take into account the multiples modalities (motor command, joints proprioception, visual stimulus, internal drives, reward, emotional states, etc.) of the brain-body-environment system. Mirror neurons are considered as a possible explanation of how social agent can imitate. Rizzolatti and al. [21] discovered in monkeys some neurons activated both when a monkey performs an action and when it observes another monkey or a human doing the same action. These mirror neurons were originally found in the area F5 of the premotor cortex. In monkeys, the mirror neuron activities are related to actions like grasping and placing objects. An observed action can produce activation even if it is partially occluded. It has been hypothesized that the role of mirror neurons is to enable the recognition of actions that are performed by others. The development of this feature would be an adaptation of evolution as it allows better social collaboration and imitation. Several models have considered this

hypothesis as a justification to action recognition capabilities and learning by imitation in robots [22]. Yet the role of mirror neurons in task learning can be questioned since it is not clear if these neurons are the cause or the consequence of skill learning. Even if some proto-imitation of tongue protrusion was observed in neonates [23], it can not be advocated that it reflects inherited imitation capabilities [24].

In our approach, we advocate that mirror neurons are a side effect of some sensorimotor learning [25], [26], [27]. In this vein, rather than concentrating on local and apparent powerful properties of mirror neurons, we will rather consider the fact that a "mirror system" spreads over several cortical structures [28]. There is evidence that the Inferior Parietal Lobule (IPL) presents some neurons with mirror pattern activation whereas in the Superior Temporal Sulcus (STS), some neurons are more sensitive to observation of actions performed by others and thus support learning by imitation features [29]. In addition, we will emphasize the role of sub-cortical structures and propose a NN model under the form of a succession of associative mechanisms processing and learning information from raw sensors to motivated decision. Hence, learning and reproducing a task is the result of a double process : first, a bottom-up process allow to assemble sensorimotor information into sequences of multimodal information (section III) . Visuomotor associations combined with torque commands allow to learn multimodal attractors that are at the basis of motor control and task learning. This PerAc loop [30], [31], before being applied to our eye-arm system, is a generalization of a pluri-disciplinary model used for interactive path learning and navigation for autonomous mobile robots.
Second, a Hippocampus-Prefrontal loop allows long term goal selection from context recognition. This contextual information induces a top-down diffusion to select the right sensorimotor transition among the possible ones (sections IV and V). Once again, this second loop was initially studied in the frame of navigation allowing a mobile robot to select efficiently the correct path in order to satisfy a given motivation. Hence, our work illustrates that a bio-inspired control architecture initially designed to allow autonomous navigation can be applied to a context of task learning in the frame of human robot interaction. Because we have followed a developmental course centered on the building of behavioral attractors based on multimodal association, the generalization of the model to the development of an eye-arm system was possible. To prepare future discussions, it is important to mention that the model is not specific to imitative behaviors. Imitation is an emergent property of the coding choices of the architecture, and there is no will to create a given mirror property.

III. Learning tasks and building attractors

In [13], [32], [33], the authors introduced the Dynamic Movement Primitives (DMP). DMP are control policies defined by a second-order dynamic system that specifies the attractor landscape for a trajectory towards a given goal. In DMP, a mixture of Gaussian kernels depending on internal variables define the changing attractor basin. One of these variables introduces a temporal reference in the movement reproduction as it converges toward an artificial attractor. The number of Gaussian kernels, their means and variances and their weights in the mixture are learned from a nonparametric regression technique of locally weighted learning [34]. This system shows interesting properties of spatial and temporal invariance and was applied to learn discrete and rhythmic movements.

In [35], a Gaussian Mixture Model (GMM) is used to encode states from proprioceptive and Cartesian information. A Gaussian Mixture Regression enables to statistically extract the adequate proprioception that is then used to control a robotic arm. Some data sets are generated during a training session and used to learn offline the parameters of the Gaussian kernels. Some works focused on making the training of the GMM more incremental [36], [37]. In [38], the controller is modified in two ways. First, Hidden Markhov Models (HMM) are used instead of GMM to encode the sequential information about the task. Then, a second order mass-spring-damping system is used to control the movement of the robotics arm. This version of the model is more similar to DMP. One of the main differences is that the learning of the constraints on the position and the velocity profile can take into account the mutual influence of different degrees of freedom whereas it is not the case for DMP. A trade-off between the control of the position and the control of the speed is managed depending on the variance on the position estimated by the model. In these works, the attraction basins are statistically learned to fit the demonstrated trajectories. Taking a different point of view, and favoring on-line learning during human robot interactions, our approach lets the attraction basin emerge from a limited number of multimodal, sensation-actions couples [30].

A. Letting sensorimotor attractors emerge from multimodal associations

A fast on-line learning based on sets of conditioning between sensory signals and motor signals is sufficient to build an attraction basin. In previous works [39], [31], [40], we demonstrated with a mobile robot (Figure 2) the learning and reproduction of a homing behavior and of a circuit following. The trajectories can be viewed as the result of a behavioral attraction basin whose shape is defined by a limited number of sensation-action couples (PerAc architecture). This principle was implemented as a neural network architecture where the sensation-action couples are the associations between the directions to be followed (action) and the place cells (states) recognized as conjunctions of visual features (landmarks) observed under specific orientations (azimuths) [39]. These associations were learned on-line through interactive teaching [31]. Moreover, we have also proposed in [40] that the "experience" of the robot during the sensation-action loop of moving along the attraction basin is the main building block of the robot's notion of place recognition. Following the same approach with a robotic arm, we propose that dynamical attractors (that is to say areas or trajectories in the working space that can be reached by the robot's arm) can be built by multimodal associations. Of course, as the robotic device has multiple Degrees of Freedom (DOF), the information is much

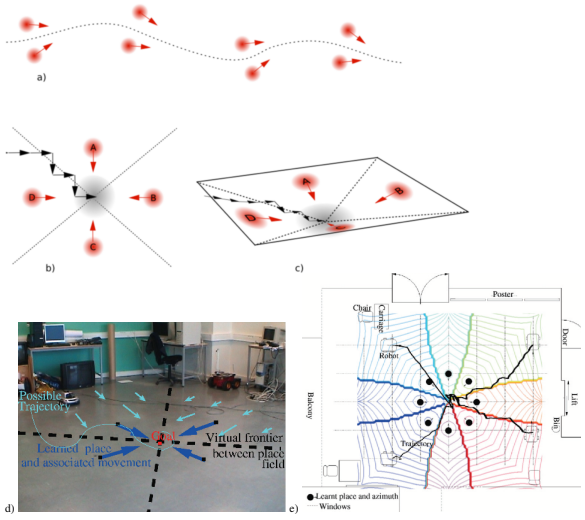


Fig. 2. Example of how is built a sensorimotor attraction basin in navigation. *a and b :)* multimodal associations can define a path (a line-shaped attraction basin) or a fixed point attractor. For example in *b,c)* we use place recognition-movement couples to built a theoretical fixed point attractor. In *d,e)* 8 places (black circles) are learned by a mobile robot at 1 m from the goal (size of the square of the floor). The theoretical place fields are superposed with the motor trajectories. The behavior of the robot is coherent with the theoretical attraction basin

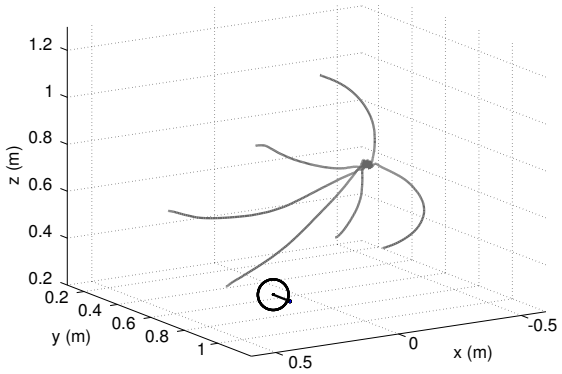


Fig. 3. Example of 6 trajectories of an arm end effector in 3D Cartesian space reaching a point of the working space and starting from 6 different initial configurations. Vision, proprioception and motor command have been associated during learning and babbling (from equations 1 to 8). As a result, S^{VM}, S^V and S^P categories allow the end effector point to converge to places that can be activated by vision or by proprioception. The black circle gives the position of the base of the robot arm.

more complex than with a mobile robot. In the following, eq. 1 and 2 describe how torque (τ) commands are generated from coefficients simulating muscle activations. Motor categories encoding proprioceptive feedback $\mathbf{p} = [\theta_\mathbf{1}, \ldots, \theta_\mathbf{N}]$ are recruited (eq. 3) and associated with visual categories to build visuomotor categories (eq. 4 to 7). These visuomotor categories are associated with the motor command coefficients $\mathbf{A}$ (eq. 8) used to generate the torque commands. As a result, a motor readout can be computed from both information (proprioception or vision), being the main basis of more complex behaviors.

We start from a motor control based on a model of muscle activation (1) implementing impedance control [41] for each joint j.

$$\forall j,\ \tau_j = A_j^+ \cdot [\theta_{j,max} - \theta_j]^+ - A_j^- \cdot [\theta_j - \theta_{j,min}]^+ - \sigma \cdot \dot{\theta_j} \quad (1)$$

The function $[x]^+$ is defined by $[x]^+ = x$ if $x \in [0,1]$, 0 if $x < 0$ and 1 if $x > 1$. The activation coefficients $\mathbf{A}^-$ and $\mathbf{A}^+$ weight the contraction of the agonist and antagonist muscles generating the movements. A damping based on velocity ensures the stability. The equation (1) corresponds to a second order damped-spring-mass like system which can be written as (2):

$$\begin{cases} \forall \text{ joint } j,\ \tau_j = K_j \cdot (\theta_j - \theta_{0,j}) - \sigma \cdot \dot{\theta_j} \\ \theta_{0,j} = \frac{A_j^+}{A_j^+ + A_j^-} \quad \text{and} \quad K_j = A_j^+ + A_j^- \end{cases} \quad (2)$$

Progressively, $\mathbf{A}^-$ and $\mathbf{A}^+$ are learned with the firsts explorations of the arm's working space and associated with visuomotor states S^{VM} (eq. 8). During the motor babbling process, the robot tries to converge toward a set of postures with continuous movements. This set can be fixed, random, or defined by the vigilance factor λ fixing during learning the overlap and the competition between the motor configurations. At the same time, proprioception is categorized (eq. 3). S_i^P are proprioceptive categories learned throughout a neural network inspired by the Adaptive Resonance Theory [42] :

$$\begin{cases} s_i^P = e^{(-\frac{\sum_j (p_j - w_{ij}^P)^2}{2\beta^P})} \text{ and } S_i^P = \frac{s_i^P}{\sum_i s_i^P} \\ Vig^P = \mathcal{H}(\lambda - \max_i(s_i^P)) \\ \Delta w_{ij}^P = Vig^P \cdot (p_j - w_{ij}^P) \end{cases} \quad (3)$$

Visual inputs S_i^V are also categorized with the same algorithm (eq. 3). When the arm is learning a configuration (learning of postures is fixed by the vigilance factor λ), S^V and S^P are associated to form visuomotor associations S^{VM}.

$$S_i^{VM} = S_i^P \cdot \sum_k (w_{ik}^{VM} \cdot S_k^V) \quad (4)$$

A confidence measure μ_{ik} for each possible connection is used to determine which connection w_{ik}^{VM} should effectively be active (5). A hysteresis effect on the choice of the active connection is produced by adding a bias h_{ik} equal to a constant h if the connection ik is the one already active ($\omega_{ik}^{VM} = 1$) and $h_{ik} = 0$ otherwise.

$$\omega_{ik}^{VM} = \delta_{kK_i} \text{ with } K_i = \underset{k}{argmax}(\mu_{ik} + h_{ik}) \quad (5)$$

The role of the confidence μ is to estimate whether a visual category should be associated with a proprioceptive configuration on the basis of the co-activation of these two categories (both being maximally active at the same time). The confidence μ in a connection is updated by (6).

$$\Delta\mu_{i_{max}k} = s_{i_{max}}^P \cdot \epsilon^\mu \cdot (\delta_{kk_{max}} - \mu_{i_{max}k}) \quad (6)$$

$$\text{with } i_{max} = \underset{i}{argmax}(S_i^P) \text{ and } k_{max} = \underset{k}{argmax}(S_k^V) \quad (7)$$

where ϵ^μ is the adaptation rate, μ_{ik} the confidence in the connection between the i^{th} proprioceptive category and the

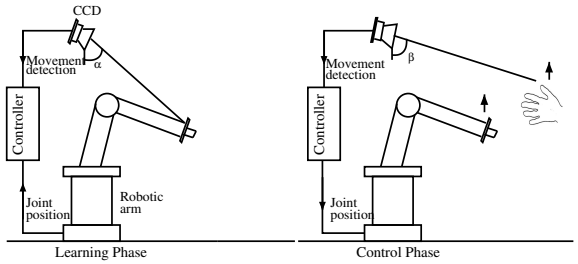


Fig. 4. *Left* : let suppose a developing controller learning from motor babbling sensorimotor equivalences between the vision and the proprioception of the arm end effector. The robot is based on elementary perception (for example movement detection) and cannot differentiate it is extremity from another moving target. Nevertheless, during babbling this elementary perception is enough to give reliable information about the end effector position since the robot is only looking at its own arm. *Right* : after learning, if someone comes in front of the robot and moves the hand, he will induce visual changes (movement detection) that the robot interprets as an unforeseen self movement. The robot acting as an homeostat, it tends to correct by producing opposite movements, inducing the following of the demonstrator gesture. Applied to an eye-arm robotic system, the generated error will induce movements of the robotic arm reproducing the moving path of the human hand: an imitative behavior emerges.

k^{th} visual category and $\delta_{ij} = 1$ when $i = j$ and 0 otherwise (Kronecker symbol). The visuomotor states S^{VM} are then associated with the activation coefficients that enables to take the posture encoded in the corresponding proprioceptive states S^P. The modification of the connective weights w^A and thus of the muscle activation $\mathbf{A} = (A_1^- \; A_1^+ \; \ldots \; A_N^- \; A_N^+)$ depends on a learning parameter α^A and the detection of a necessary correction C_j for the j^{th} joint. The motor controller is adapted by increasing the muscle activations to contract the muscles so that the arm moves in the correct direction. The second term in the learning rule enables to decrease the weights of the connections to avoid saturation of the weights.

$$\left\{ \begin{array}{rl} A_j = & \sum_k w_{jk}^A \cdot S_k^{VM} \\ \Delta w_{jk}^A = & \alpha^A \cdot C_j \cdot S_k^{VM} \cdot (1 - w_{jk}^A) \\ & -\gamma \cdot w_{jk}^A \cdot \underset{j}{max}[A_j^+ + A_j^- - 0.1]^+ \end{array} \right. \quad (8)$$

Depending on the importance given to visual information, the visuomotor states S^{VM} can be selected by the visual input thus control the movements to maximize the recognition of this visuomotor state by moving to the corresponding proprioceptive configuration (see Figure 3). The overall dynamics resulting from eq. 2, 3 and 8 can be compared to the Neural Fields motor control used in behavior based systems [43] (we also used DNF in previous works on imitation [44]). Nevertheless, in our solution, the computing cost is lower than the Neural Field, this latter depending on Amari equations [45].

B. From sensorimotor association learning to imitation

Spontaneous immediate imitation of simple gestures, often designed under the terms of "emulation", "response facilitation" or "low-level imitation" and identified as early imitative behaviors [3], [6], can be seen as a side effect of our control architecture. Two principles are at the origin of this imitative behavior : the ambiguity of the perception, and the S^{VM} associations that force the robot to maintain an equilibrium between S^V and S^P, that is to say to act as an homeostat [44]. If a human comes in front of the robot after S^V, S^P and S^{VM} categories have been learned, and if the robot's perception do not discriminate the human's hand from its own arm end effector, then spontaneous imitation emerges. Elementary perception such as color or movement detection is enough to activate visual inputs S^V and the sensorimotor category S^{VM}. The activation of S^{VM} then triggers the associated muscle activations $\mathbf{A}$. As a consequence, moving the hand in front of the robot will induce visual changes that the robot interprets as an unforeseen self movement. The perception-action control loop from S^V to the activations $\mathbf{A}$ acts as an homeostat and tends to maintain the equilibrium (corresponding to the associations learned during babbling) between visual and proprioceptive information by generating adequate torque commands. Therefore, the robot moves its arm so that its proprioceptive configuration changes to the S^P category associated to the perceived visual stimuli S^V via S^{VM}. This correction induces the following of the demonstrator's gestures by the robotic arm, and a behavior of gesture imitation emerges (Fig. 5). Interestingly, this immediate imitation is obtained by a robotic controller that has no a priori model or notion of "other". It is simply regulating by acting the unbalances of its visual and motor perception. Hence, only the visual consequences of the actions are imitated. Of course, it means that the movements are reproduced with a reduction of movement amplitude due to the change in the visual perspective. We can see this in Figure 6, where two robots with the same arm (4 controlled Degrees of Freedom (DOF) and a pan-tilt camera) are settled in an imitation situation. The robot on the right hand side performs a given sequence of gestures. The robot on the left hand side is controlled by our model. It focuses its attention on the hand of the demonstrator robot believing it is its own hand[1].

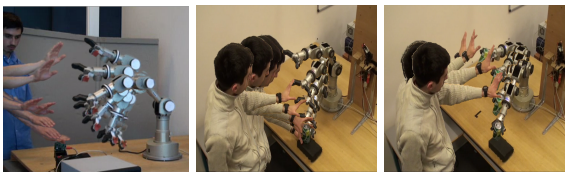


Fig. 5. *Left* : Robotic transposition of the setup depicted at Figure 4. Qualitative comparison of imitated gestures performed in front of the robot. Perception ambiguity and an homeostatic controller induce movements to maintain a perceptual equilibrium. The robot performs low level imitation of directly observed gestures. *Middle and right* : Gesture imitation can be used to bring the robot's end-effector toward objects (here, to grasp a can) or interesting part of the environment, and becomes the common basis of learning by observation and learning by doing.

Other works have focused on imitating the posture rather than the visual consequences [46], [22], posing the correspondence problem [47] as a crucial issue of human-robot

[1]Here, a bias is introduced by removing the original color (red) of the gripper of the imitator robot. At first, the robot could focus either on its own hand or on the mistaken hand beginning to imitate or not. The issue is then where the attention of the robot is. Interestingly, if there is almost synchrony, there is no competition and the center of attention is shifted to the merging of the inputs inducing the imitation.

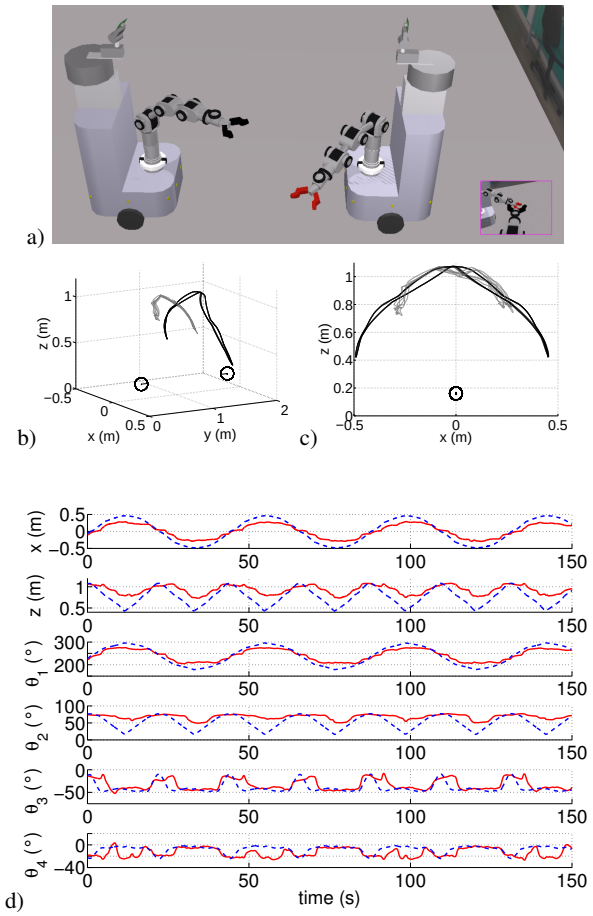


Fig. 6. Low level imitation of gestures based on observed visual effects. The robot on the left (a) imitates the moving robot on the right. The trajectories in the Cartesian space are plotted in (b,c) in gray for the imitator and black for the imitated. As only the visual consequences of the gestures are imitated, the imitator robot (red plain line) reproduces the gestures of the imitated robot (dash-blue line) with less amplitude as well as a slightly delayed phase. These effects can be seen both in the absolute Cartesian position of the arm end effector and in the angular values of the controlled joints (d).

imitation. At the opposite, we wish to show that our very limited controller is able to initiate imitations without taking into account any correspondence or human-robot equivalences. Such behavior is very important, since it engages the robot in a gestural interaction. If this interaction is close enough, that is to say if the human's hand is at the "surface" or inside the robot's workspace, the amplitude reduction is negligible, and the imitation becomes an opportunity to provokes moves of the robot toward objects (Fig. 5), or interesting parts of the environment. This spontaneous reproduction of meaningless gestures becomes the basis of task learning by observation.

IV. Learning by observation

Piaget has identified the apparition of deferred imitation, also called learning by observation or observational learning, around the age of 18 months [10]. Deferred imitation is the ability of the observer to reproduce an action that was previously demonstrated by a teacher. Most of the time, the observer does not produce movements during the observation phase, and only the demonstrator is performing the action. According to Piaget and numerous authors in developmental psychology, the apparition of deferred imitation is an important landmark in development, revealing the ability to connect observations with internal symbols, memory and action. It is also the first apparition of the powerful learning function of imitation. As a result, the models and interpretations accounting for learning by observation have often been distinguished from low-level "mimicries" such as emulation or early spontaneous imitation of gestures. In our understanding of imitation behaviors, our model tends to show that these two kinds of imitation should not be considered as totally independent as they share numerous common mechanisms. Indeed, in our model, inhibition of own actions (to observe instead of acting) and a memory of what was observed is enough to shift from low-level imitation to learning by observation.

In order to be able to store the "elements" that constitute an action, which can be observed as well as performed by our robot, we have added a model of the Hippocampus that allows to learn a sequence of elements (the central layer in Figure 1). The role of this network is to learn and predict the changes in the sensorimotor events. The network is inspired by the functions of two brain structures involved in memory and timing learning: the cerebellum and the Hippocampus (see [48] for the neurobiological model we have proposed). For sake of simplicity, we will not describe the whole model in this paper, and readers can refer to [48], [49], [50], [51], [52] to have a complete neurobiological description of the network, with related experiments in the frame of navigation or timing prediction. To summarize, this model of the Hippocampus enables the robot to learn and reproduce sequences of elements $E_1, E_2, .., E_n$ under the form of transitions between the current element and a time trace (short term memory) of the previous element. To recall a sequence, the system just needs to be fed with the first element E_1 of the sequence. The recognition of E_1 will activate the transition $E_1 - E_2$ that will in turn activate E_2. The recognition of E_2 will activate the transition $E_2 - E_3$, and so on. The timing of the transitions can also be learned by the network [51], [52], with predictions that preserves this timing with an accuracy proportional to the inter-stimulus interval.

In the case of our arm control architecture, the Hippocampus model is able to learn the sequence of categories S^V or S^P activated during the behaviors of the robot. For example, a sequence of gestures can be taught to our robot via immediate imitation by letting the Hippocampus memorize the sequence of S^P that have been activated during the robot's moves. Similarly, from the architectural point of view, a switch to deferred imitation is easy : we can let the Hippocampus learn the sequence of visual categories S^V activated by the teacher's hand during the demonstration. During this demonstration, the robot must also inhibit its own motor outputs (Figure 7). Then, the reproduction can be done as the robot internally rehearses the sequence of transitions between the S^V. Each prediction of a S^V activates the related S^{VM} category that triggers

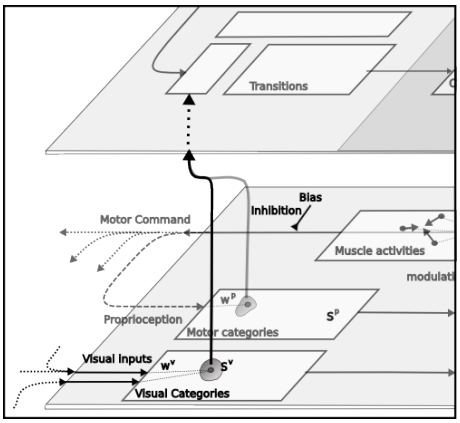


Fig. 7. Changes in the model to switch from immediate imitation to deferred imitation. An internal bias allows to inhibit motor actions when the robot observes the demonstrator. This bias is an ad-hoc parameter, whose control should be linked to parameters depending on the dynamics of the interaction (this point is out of the scope of the paper, and we discuss it in section VI-A). S^P categories are also inhibited, and only S^V activate the transition prediction for sequence learning. As a result, ambiguous perception allows to learn the sequence of S^V activated by the demonstrator's hand, and the associations S^{VM} to S^P allow the consecutive reproduction when the inhibition bias is removed.

a motor readout and the corresponding gesture. The robot reproduces the demonstrated sequence of gestures according to what was perceived during the demonstration. Hence, the robot is capable of some deferred imitation [53] by alternating motor inhibition during observation and motor activation during reproduction of the encoded sequence (visual or proprioceptive). Figure 8 presents an experiment where the robot learns to catch and move a can. As in previous section, we use color detection to activate S^V and play with the ambiguity of perception as the skin tan of the demonstrator's hand will activate the S^V on the visual path of the demonstration. The task is composed of four S^V. The number of categories is only dependent on the density of S^V and S^{VM} learned during babbling, and such visuomotor "resolution" of the working space can be refined during the development of the robot. The first category is the visual position of the hand of the demonstrator before he catches the can, the second one is the visual position of the hand when the teacher catches the object, the third one is an intermediary point on the trajectory and last state is the position of the hand when the experimenter releases the object. After learning, motor inhibition is turned off, and the rehearsal of the sequence will be possible if the first category S_1^V of the sequence is activated. In this case, the arm will try to reach the proprioceptive configuration which best fits to the predicted S_2^V according to S^{VM}. Finally, the robot reproduces a reaching and grasping (reflex[2]) of the object similar to the one performed by the human partner. It then moves it to the final position following the same trajectory as in demonstration.

[2]Grasping issues were strongly simplified, using an ad-hoc procedure driving the robot to close its gripper whenever an object is detected by the proximetric sensors of the fingers. The orientation of the grip was also always horizontal in our case.

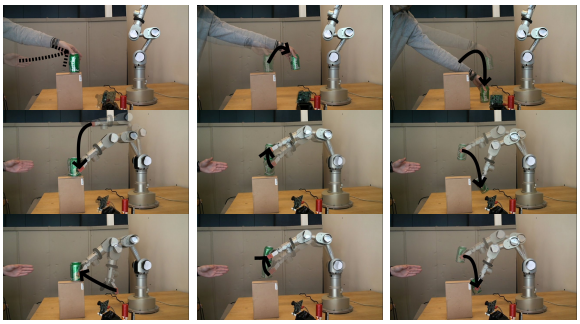


Fig. 8. Deferred imitation based on visual sequential learning : an emergent cooperative behavior. The robot is shown the action : grasping and moving an object to a given location. In each picture, two successive positions of the arm are superposed with the black arrow showing the performed movement. Up : Learning phase. The robot sees the teacher doing the action. Middle and Bottom : Reproduction of the action from different initial position. The robot sees the visual trigger from the hand on left side of the visual field. It then completes the sequence previously demonstrated according to its motor capacities.

Our setup is comparable with what is done in the learning from demonstration paradigm [54], [55], [56]. During the demonstration phase, the robot learns the actions to be done while staying still (motor inhibition). Then, the robot can use the newly acquired sequence to reproduce the demonstrated task. In [55] and [56], the task learning is based on a model training which needs several sets of data provided by the demonstration. Besides, this training is usually based on proprio-motor coding. When using vision to recognize and monitor the task, an a priori inverse model is usually used. On the contrary, our architecture uses one-shot learning of the sequence. Our model can then learn the action directly in the visual space because it has previously learned the inverse model of its arm in the workspace during the babbling with the S^{VM} and $\mathbf{A}$ associations. No model of the task was a priori provided by a human. So our robot can perform more autonomous - though rougher - deferred imitation based on observation.

Our robot reproduces a sequence of known gestures without specific meaning. Whereas the robot has no a priori representation of the task to be done, the sequence drives the robot to catch an object and move it to another place. The robot only discovers that there is an object when it closes its gripper on it and starts to move it. The interaction with the teacher can help not only to learn new tasks but also to discover new possible interactions with the environment [57]. Due to the position of the camera in the experiment, the robot does not try to reach the experimenter's hand because its position is outside the robot working space. To an external observer, the human's gesture shows his intentions (here the hand pointing to the object) and the robot's behavior in response (grasping the object) can look like it recognizes the human's grasping intention and tries to cooperate performing the grasping. This is an emergent property that comes from the interplay between what the robot can and cannot do and the ambiguity of its perception.

V. TOWARD ACTION PLANNING

We have shown how simple categorization and associations can be used to build a bottom up representation of sequences of gestures to accomplish tasks. Though, with only these capabilities, the robot would not be able to store more than one sequence in memory and moreover to select adequate sequences depending on situations. In order to extend the model, information descending from external contexts and internal motivations should come down to enable a correct action selection.

A. Encoding sequences with topological maps

In animat approach, the issue of navigating toward resource places introduces the need for associating drives with the action selection part of the system. In [58], [59], a robot or an animal finds resources in its environment and learns how to return to them when needed. To each type of resource corresponds a drive that represents the motivation of the robot to look for that kind of reward (e.g. for an animal, a drive can be hunger and the reward eating food). Models based on reinforcement learning can explain the adaptation and selection of action in order to get rewards [60], [61]. A drawback of reinforcement learning is the convergence time that is usually long as the potential value must be back-propagated between the different actions. An alternate solution is a latent learning of a topological model of possible actions called cognitive map. In navigation, the cognitive map can be a topological graph of transitions between place-cells that can explain the planning capabilities of an animal and the fast learning with only a few rewards [59]. The learning of the cognitive maps can happen before the appearance of any reward. Thus, they can be used rapidly to plan the shortest path to particular locations (goals) associated with rewards (when they are finally given). Besides, if an obstacle prevents the agent from following one particular path, the agent can replan its trajectory immediately. Cognitive maps were used in robotic navigation [59]. In the present adaptation of a cognitive map to arm control, it will be used for its property of learning and goal managing depending on motivational contexts. Whereas, in animat approach, reward and drives were linked to physiological needs, in our experiment, we assume that motivational contexts could be categorized and learned and that a social "reward like" feedback could be conveyed when the robot performed well. The possible transitions between states are learned when the arm moves from one state to the next. This is done in the sequence learning part of the architecture. The learned transitions are included in the graph of the different transitions (cognitive map). It encodes the information about possible paths by linking consecutive transitions.

Upon reception of a reward ($R = 1$), the neuron coding the last performed transition in the cognitive map is associated to the most active motivational context. Performing this transition in this context is then a new goal for the system. The associations between the motivational contexts (drives) M_j and the neurons O_i corresponding to transitions in the cognitive map are stored on the connective weights w^{motiv} with the following learning equation:

$$\Delta W_{ij}^{motiv} = R \cdot M_j \cdot (\alpha \cdot (1 - W_{ij}^{motiv}) \cdot O_i - \gamma) \quad (9)$$

where α is a learning rate, γ a decay rate and R is the reward signal (0 or 1). When a reward is received ($R = 1$), the activity O_{i_m} of the neuron corresponding to the last performed transition is set to 1 and others neural activities ($O_{i \neq i_m}$) in the cognitive map are set to 0. Thus, the connection from the active motivational context M to the neuron O_{i_m} (last performed transition) is reinforced whereas the other connections are decayed.

When a context is active, its activity is back propagated in the map between consecutive transitions with weights $w < 1$. As a result, transition neurons in the cognitive map have an activity proportional to the goal-transition activity and the distance (in term of number of transitions) to it (see [59] for more details). In a given state, different transitions can be realized. The corresponding activities in the cognitive map can be used to bias the selection of action. Doing so, the system performs a gradient ascent toward the most interesting goal (close, with high reward potential) depending on the active motivational context.

B. Robotic experiment

Fig. 9. Learning a pick-and-place task with a cognitive map. The human teacher demonstrates how to grasp an object through passive manipulation of the arm. The first object is a red can. When the infrared sensors on the gripper detect the presence of an object in its range, a grasping reflex is triggered. Force sensors on the gripper then detect when an object is held. The state of the gripper is then open or closed. The arm is manipulated again and brought above one of the dropping boxes, learning a trajectory between the picking place and the dropping box. Due to mechanical limitations, the interacting human can use a button located on the gripper to command the opening of the gripper. Similar protocol enables to learn the task associated with the green can. Using the encoded cognitive map, the robot reproduces alone the sequences that where demonstrated depending on the color of the can.

The model of cognitive map for arm control was implemented on a 4 DOF Katana robotic arm [62]. The task is a pick-and-place task learned in the proprioceptive space from passive learning by manipulating the robot arm (see Fig. 9 for the robotic setup). At first, the robot has no knowledge about its environment but the recognition of contexts and positive feedback given by the teacher (reward). We use a simple visual categorization system. The recognition of contexts and social reward is assumed to be already available in the system. Though it should have been learned, here it is a priori given by visual feature extraction (e.g. color). The robot can discriminate two different motivational contexts corresponding to two types of objects (red or green). The motivational context of the robot results from a competition with hysteresis on the recognition of the two possible contexts. During the experiment, an object (a colored can) is first presented in the

reaching space of the robot at a given location (i.e. the picking place). The robot must grasp the can and drop it into one of two boxes, also at static positions in the reaching space. A monocular camera is used as the visual system of the robot, focusing on the picking place. We assume that the visual attention of the robot is always directed toward that place. The human teacher demonstrates how to grasp an object through passive manipulation of the arm. The first object is a red can. During the movements, the succession of activated S^P induces the learning of transitions with the same process as in previous section. Meanwhile, the cognitive map encodes these particular sequences of transitions in its topological graph. When the can is dropped in the correct box, a positive feedback (reward like signal) is given to the robot which learns the association between the active "red can" context and the last performed transition i.e. the goal of releasing object at this position. The different learning and reproduction phases are shown in Figure 10. At each time, the most active context propagates an activity to its associated goals. The activities are diffused to the previous transitions in the cognitive map. Figure 11a gives an example of the propagated activities of the transition neurons when the goal "to box 2" is active. The cognitive map encodes the successive transitions represented by the arrows whose size and color depend on the diffused potentials of the transitions. Transitions close to the goal have high potential so are represented by big black arrows. Depending on the current state of the robot, the gradient to be followed is given by the propagated activities. A simple comparison between the potential activities of the current possible transitions enables to select the next transition to do. As in previous section, the selected transition then gives the predicted next S^P, and motor read-out is obtained by the related association between S^{VM} and the activation coefficients $\mathbf{A}$ defining the torque command.

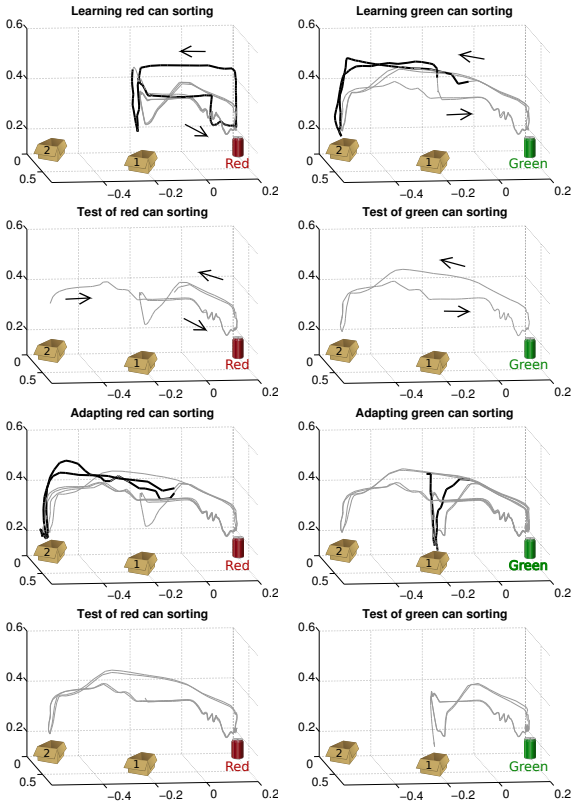


Fig. 10. Trajectory of the arm end effector in the 3D Cartesian space. The experiment can be explained as different phases, each one corresponding to a row. The first phase is the learning of the pick-and-place task for the red can and the green can. The black thick line corresponds to the trajectory during passive manipulation of the arm. The second phase is the validation of the reproduction of the task after learning. The third phase is the adaptation of the learning to new conditions: the reversal of the boxes for red and green can. The movement of the arm is corrected using passive manipulation to show the expected gestures. The learning of the new conditions needs several reinforced corrections to counterbalance the initial reinforcement of the first condition. The last row shows the adapted reproduction after relearning.

C. Properties of the controller

Figure 11 shows the position of the learned states in the workspace and the transitions learned between states at the end of the task. Interestingly, more states were recruited near the picking place than for the movement between the boxes. During the demonstration of the grasping, the different proximo-distal articulations were manipulated to reach the object with accuracy and with the correct orientation for the gripper. As the states encode the proprioceptive information in the articular space, they can be close in the 3D Cartesian space but distant in the multi-dimensional joint space of the arm. On the contrary, during the demonstration of the simplified left/right and up/down movements to the boxes, only the proximal joints were manipulated as there were no particular constraints to be shown for the other joints. As a result, the varying density of states is explained by the recruitment process depending on each joint variation from the encoded joint/gripper configuration (sec. III-A).

Figure 11b displays both the position of the encoded proprioceptive categories and the reproduced trajectory of the arm end effector in the 3D Cartesian space. For each point of the trajectory, the color corresponds to the color of the maximally recognized category. Due to its response activity in the neighboring area, a category can be recognized before the arm end effector is in the exact encoded posture. The system consequently anticipates the next move without finishing the current one, which leads to small shortcuts being taken while reproducing the trajectories, in spite of the demonstrated ample movements. This effect is more visible when there are only a few categories like in the middle of the trajectory from the picking place to the boxes. Less constraints during demonstration implies less states and thus more shortcuts in the reproduced movements.

As the environment changes, the context and/or the current state of the robot may also change. Re-planning is quite fast as the propagation of the new potential activities (context changed) in the cognitive map only requires a few iterations to be updated depending on the longest path in the cognitive map. Otherwise, if only the current state changed, it immediately predicts other transitions and the selection bias is read out in another part of the cognitive map. When the task changes

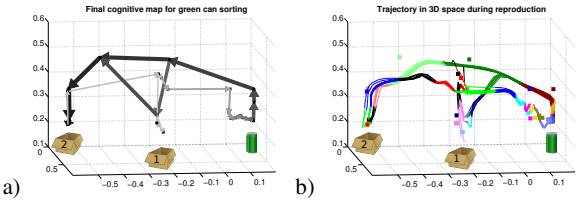


Fig. 11. Encoded states and trajectories during experiment. a) Encoded cognitive map. Squares are the states. The sizes and colors of the arrows correspond to the activities of the neurons in the cognitive map associated to the corresponding transitions. b) The color of the trajectory correspond to the recognized state at the time.

and the behavior must be adapted, instead or relearning the whole motor sequence, a fast re-association of the transitions with correct contexts enables to adapt the goals and the reproduced sequences. The already learned motor sequences can be reused.

VI. Discussion

A system unable to discriminate visual information about self movements from the movements of others can be seen as very limited from a classical engineering point of view. However, in our approach, the perception ambiguity is a key phenomenon allowing to bootstrap the development of various imitating behaviors with an autonomous robot. Whereas this hypothesis can be seen as very speculative, numerous psychological works show comparable human behaviors when visual perception is ambiguous. In 1963, Nielsen [63] proposed an experiment in which subjects were placed in front of a semi-reflecting mirror (the experiment was replicated more recently by Jeannerod and colleagues [64]). In a first condition the mirror is transparent, and the subject sees his own hand placed on a table under the mirror. In a second condition the mirror reflects, and the subject sees another hand (the demonstrator's hand) that he will mismatch for his own hand. Because a black glove has been put on both hands, the subject has the feeling to see his own hand and does not imagine there is another hand in the experiment. During each trial, the subject has to draw a straight line with a pen in the direction of his own body axis. When the perceived hand is his own hand, the performance is perfect. When the mirror reflects, the subject "imitates" the other hand movements and do not perceive any difference if both hands are almost synchronous. If the perceived hand moves in a quite different direction, the subjects tend to correct the error by drawing in the opposite direction but they never suspect the presence of another arm (they believe that the "wrong" trajectory is due to their own mistake).

A. Our model and the double function of imitation

Spontaneous imitation of simple movements, emulation, or response facilitation can all be seen as behaviors emerging from the conjunction of ambiguous perception and homeostasis, that is to say the tendency to regulate unbalanced perception by acting on the environment. The unbalance is caused by the ambiguity, and the responding action is the imitation behavior (Fig. 4). In section III-B we proposed a robotic transposition, similar from the setup of Nielsen and later Jeannerod. We show that the low-level imitation behavior emerging from our controller becomes a starting point for discussing the powerful learning function of imitation with observational learning. As a result, we show that our control architecture, with very limited perception is a good basis for task learning. In section IV, learning by observation is possible in dyadic (human-robot) situation. In section V, learning, adapting and planning multiple complex tasks is also possible in triadic (human-object-robot) situations. Interestingly, all these tasks have been done with an embedded system that has almost no a priori information about caregivers, the objects, and the environment. Or course, important limitations still exists in our model : we have to trigger the motor inhibition during observation and also we decide the nature of the information (S^P *vs* S^V) that activates the transition. We still have to address the problem of information selection (or information inhibition) between S^P, S^V and the torque command.

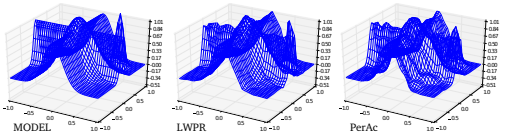
Our recent works suggest that looking toward the communication function of imitation may be the beginning of a solution. Developmental psychologists have underlined how early games based on low-level imitation often have a social and communication purpose [65], [6]. In this case, the goal is not to learn a new task, but rather to be able to bootstrap and entertain a non-verbal interaction based on exchange of gestures with the pleasure to mirror and to be "inside" the interaction. We hypothesize that simple imitation games between two agents can be bootstrapped by perception ambiguity and then become active processes by which the robot starts to characterize its own sensorimotor dynamics [66]. Particular perception-action dynamics such as synchronous [67] , rhythmic [68] or contingent [7] exchanges could help to select between S^P (my own proprioceptive dynamics drives) and S^V (visual dynamics drives) and to inhibit motor outputs. Hence, finding the right information selection should lead to the establishment of turn taking or role switching and allow to progressively build a bound between self and other. Imitation is the link toward the share and comparison of these multimodal attractors, and an opening to synchrony, phase lock, turn-taking and "like-me-ness" detection mechanisms.

B. Accuracy and generalization

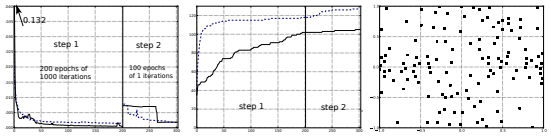
The accuracy of the experiments presented in previous sections depends on the number of S^{VM} associations (linked to the number of categories S^P and S^V). Our model does not optimize the number of states and the shape of the encoded categories. Though, with enough categories, it can perform correctly in comparison to the on-line regression techniques like LWPR (Table I). The accuracy can be increased by using adaptation of the movements so that the system can reach positions that were not learned. The method of Fukuyori et al. [70] is interesting because it relies on preexisting attractors

TABLE I
COMPARISON WITH LOCALLY WEIGHTED PROJECTION REGRESSION

Locally Weighted Projection Regression (LWPR) is an algorithm that can solve regression problems and was applied to motor control (see [69] for details). Our model (PerAc [30], [31], Appendix) is based on simple categorizations and association between sensory input and motor output. We compared both models in a classical test of function learning. The test was made as follow : in the first step, LWPR and PerAc have to approximate a function based on the combination of 3 Gaussian. In a second step, at time $t = 200$ epochs (20000 iterations), the cross function is modified by adding a narrow Gaussian kernel in the upper-right quadrant. Both algorithms have then to re-adapt their basis to fit to the modified landscape. Results are shown in the following figures :



After 20000 learning iterations (end of step 1), both models performs equally well : LWPR slightly outperforms Perac with mean square errors (mse) respectively equal to $4.94e^{-4}$ and $1.5e^{-3}$. The predicted functions are similar to the model function, with a smoother result for LWPR.



At the end of step 1, LWPR and PerAc models have recruited a comparable number of fields (categories) : respectively 102 and 118. PerAc recruits noticeably faster, suggesting a faster approximation of the whole shape of the model. During step 2, the PerAc model adapts quicker and thus performs better during the first 50 iterations. This result is consistent with the use of PerAc model for on-line learning, and situations where the robot should rapidly provide satisfying behaviors with a limited number of (time consuming) demonstrations. In the long run, LWPR fills the gap with PerAc performing similarly to better. Besides, LWPR can learn projections to optimize the encoding when data are in low dimensional manifolds. The main elements of the comparison are summarized below :

	LWPR model	PerAc model
Principle	Local linear models associated with recruited and adapted Receptive Fields	Categorization of input recruited on the basis of activity versus vigilance and also high output error
Learning	Scale of Receptive Field, linear model	Recruitment of sensory categories, association with motor output based on conditioning
Advantages	• regression of non linear functions on high dimensional input space (but with data in low dimensional many fold) • *model using learned projections in lower dimension space to avoid the dimensionality curse*	• Fast on-line learning from corrective and interactive teaching • adaptation of the inputs encoded in categories • Simple model with less computation to learn and encode the prediction
Drawbacks	requires enough data to update correctly both the Receptive Fields and the linear models	• no adaptation of selectivity of the categories; • no smoothness constraint on learning

to explore a narrow part of the space and converge to a target. Depending on a feedback (e.g. visual distance between target and current position of arm end effector), the system can evaluate its performance and switch between exploiting the learned visuomotor attractors or exploring to reach a better position [53]. Of course, such solution implies that the robot reached a developmental stage where it can discriminate objects and targets in its environment.

The experiment presented in Section V-B was encoded in the proprioceptive space. This task can also be encoded in the visual space (using visual states like in Section IV). Finally both encoding should be processed in parallel to let the robot choose in which space the task is performed. Picking the can in different positions may imply to use contexts encoding the position of the can in the visual space thus activating different trajectories. The use of a priori known contexts with planning model also eludes the difficulty of learning the adequate contexts. While receiving many multimodal inputs, the system should learn to extract the relevant features discriminating the situations and thus enabling it to adapt its behavior. The theory of chunking [71], [72] describes the use of states representing a particular piece of information sufficient to perform some tasks. Future work is to study how such chunks can be learned, encoded and used in the model at the level of the contexts but also of the states used to code the transitions.

C. Social interaction to improve cooperation

After learning sequences of gestures from imitation (visual or passive, direct or observed), the system should be able to reuse already learned sequences if they can fit. In order to improve cooperation, an interacting human should progressively be able to provide a feedback to guide the behavior of the robot without needing to demonstrate again and again the desired motor behaviors to the robot. The feedback signal can be linked to verbal and non-verbal social cues [11]. A robot head that can display and recognize facial expressions could be added to the robotic setup of this experiment to make the interaction between human and robots easier. With such as system, the human could directly give reinforcing or punishing signals through interaction, by displaying positive or negative emotions. This kind of interaction has already successfully been used for the social referencing of objects with good performances [73]. In order to adapt this enhanced interaction capabilities in our setup, the use of negative feedback must be defined. A negative feedback can prohibit either the current movement or the current goal perceived by an observing human. With the cognitive map architecture, the local gradient reading of the propagated activities do not let the robot explicitly manage its own goals. An extension of the model was detailed in [74] to allow such managing. It is based on testing potential goals and evaluating if they could be the goal producing the currently followed gradient. This extension could effectively improve the sorting pick-and-place task by enabling to signal that a goal is wrong. Therefore the robot could adapt its behavior without any tedious interaction from the teacher. The facial expression and direction of gaze of the teacher can also provide the necessary cues for guiding

the attention of the visual system to interesting objects [75] and help categorize them. The robot would then be able to learn from interaction with a human partner which objects are interesting and which kinds of object should be categorized into different contexts in order to select the correct behavior in any situation.

APPENDIX

DESCRIPTION OF THE PERAC MODEL USED IN THE COMPARISON WITH LWPR MODEL

The comparison between LWPR model and PerAc model was done with the example provided with the downloadable library of LWPR [76]. The test consists in learning an approximation of a simple non-linear 2D (x_1, x_2) to 1D (y) cross-like function (eq. 10).

$$\begin{cases} a = e^{-10 \cdot x_1^2}\ ,\ b = e^{-50 \cdot x_2^2} \text{ and } c = 1.25 \cdot e^{-5 \cdot (x_1^2 + x_2^2)} \\ y = max(a, b, c) \end{cases} \tag{10}$$

The learning is performed during 200 epochs of 100 iterations. The performance of each model is evaluated at the end of each epoch. After these 200 epochs of 100 iterations, another 100 epochs of only 1 iterations are computed so that the model learns the modified cross function (eq. 11) which is the original cross with a narrow peak in $(0.6, 0.6)$.

$$\begin{cases} a = e^{-10 \cdot x_1^2}\ ,\ b = e^{-50 \cdot x_2^2} \text{ and } c = 1.25 \cdot e^{-5 \cdot (x_1^2 + x_2^2)} \\ d = 1.25 \cdot e^{-75 * ((x_1 - 0.6)^2 + (x_2 - 0.6)^2)} \\ y = max(a, b, c, d) \end{cases} \tag{11}$$

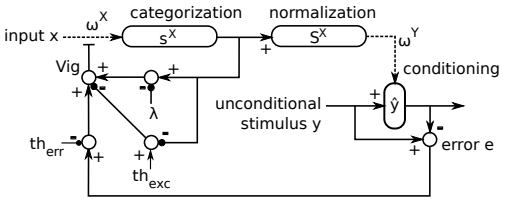


Fig. 12. PerAc architecture (simple association and conditioning learning) with recruitment depending on both input (x) recognition and output (y) error.

The PerAc model relies on a categorization process and a conditioning process (Fig. 12). Categories s^X are recruited (eq. 12) to encode the input x in a process similar to the recruitment of proprioceptive states (eq. 3) described in section III-A. So, s_i^X are also categories learned throughout a neural network inspired by the Adaptive Resonance Theory [42]. The main difference between the two sets of equations is the recruitment signal Vig can be active not only when the maximal recognition activity is under the vigilance threshold λ but also when the output error e (eq. 14) is over an error threshold th_{err}. However, if there is already a category coding a similar pattern (maximal activity over an exclusion threshold th_{exc}), no new category will be recruited. A new recruited input pattern is encoded on the synaptic weights $\omega_{i'j}^X$ of the neuron i', which was not already recruited.

$$\begin{cases} s_i^X = e^{(-\frac{\sum_j (x_j - \omega_{ij}^X)^2}{2\beta^X})} \text{ and } & S_i^X = \frac{s_i^X}{\sum_i s_i^X} \\ Vig = \mathcal{H}(\lambda - \max_i(s_i^X)) & +\mathcal{H}(e - th_{err}) \\ & -\mathcal{H}(\max_i(s_i^X) - th_{exc}) \\ \Delta\omega_{i'j}^X = Vig \cdot (x_j - \omega_{i'j}^X) \end{cases} \tag{12}$$

The activities of the categories are normalized (S^X). The learning of the output to be associated to these categories is based on Widrow&Hoff rule [77] performing conditioning learning as a simple gradient descent. The predicted output $\hat{y}$ is computed and compared to the desired output y (eq. 13). The synaptic weights ω_i^Y are adapted according to the error of prediction. When there is recruitment ($Vig = 1$), the learning rate ϵ_N is high to enable a fast encoding of the new pattern. The smaller constant learning factor ϵ enables the adaptation of already encoded patterns. The learning rate ϵ_N also depends on the norm of the input S^X to the conditioning learning layer. As a result of this variable learning rate, the learning is fast when there is (almost) only one active neuron in S^X and slightly slower when there are several neurons activated at the same time. This property enables a better averaging learning when there are interferences between the learned categories. It avoids too big oscillations of the connection weights during the learning.

$$\begin{cases} \hat{y} = \sum_i \omega_i^Y \cdot S_i^X \\ \Delta\omega_i^Y = \epsilon_N \cdot S_i^X \cdot (y - \hat{y}) \\ \epsilon_N = \frac{max(\epsilon, Vig)}{||S^X||} \end{cases} \tag{13}$$

The error between the predicted output and the desired output is simply the Euclidian distance between the two vectors (though output y is only 1 dimensional in this test) :

$$e = sqrt((y - \hat{y})^2) \tag{14}$$

The PerAc algorithm can rapidly recruit new categories when there are errors on the predicted output. Categorizing these particular situations will enable to retrieve the correct responses faster which is useful when learning on-line in situation of interaction. It can also adapt the prediction without recruiting in order to avoid to use too many categories to code the same area of the state space.

ACKNOWLEDGMENT

This work was supported by the INTERACT French project ANR-09-CORD-014, the French EquipeX ROBOTEX, and the French TINO and SESAME projects.

REFERENCES

[1] M. Tomasello, M. Carpenter, J. Call, T. Behne, and H. Moll, "Understanding and sharing intentions: the origins of cultural cognition." *Behavioral and Brain Sciences*, vol. 28, no. 5, pp. 675–691; discussion 691–735, 2005.

[2] K. W. Spence, "Experimental studies of learning and higher mental processes in infra-human primates." *Psychological Bulletin*, vol. 34, pp. 806–50, 1937.
[3] A. Whiten and R. Ham, "On the nature and evolution of imitation in the animal kingdom: Reappraisal of a century of research," *Advances in the study of behavior*, vol. 21, 1992.
[4] M. Tomasello, M. Davis-Dasilva, L. Camak, and K. Bard, "Observational learning of tool-use by young chimpanzees," *Human Evolution*, vol. 2, no. 2, pp. 175–183, 1987.
[5] R. Byrne and A. Russon, "Learning by imitation: a hierarchical approach," *Behavioral and Brain Science*, vol. 21, pp. 667–721, 1998.
[6] I. C. Uzgiris, "Imitation as activity: its developmental aspects," in *Imitation in infancy*, J. Nadel and G. Butterworth, Eds., 1999, pp. 187–206.
[7] J. Nadel, K. Prepin, and M. Okanda, "Experiencing contigency and agency : first step toward self-understanding ?" *Interaction Studies*, vol. 2, pp. 447–462, 2005.
[8] P. Andry, P. Gaussier, S. Moga, J. Banquet, and J. Nadel, "The dynamics of imitation processes: from temporal sequence learning to implicit reward communication," *IEEE Trans. on Man, Systems and Cybernetics Part A: Systems and humans*, vol. 31, no. 5, pp. 431–442, 2001.
[9] E. L. Thorndike, "Animal intelligence. an experimental study of the associative processes in animal," *Psychological Monographs*, vol. 2, 1898, (4, whole No.8).
[10] J. Piaget, "La formalisation du symbole chez l'enfant. imitation, jeu et rêve. image et reprsentation," 1945.
[11] P. Zukow-Goldring and M. A. Arbib, "Affordances, effectivities, and assisted imitation: Caregivers and the directing of attention," *Neurocomputing*, vol. 70, no. 1315, pp. 2181 – 2193, 2007.
[12] P. Jacob and M. Jeannerod, "The motor theory of social cognition: a critique." *Trends in Cognitive Sciences*, no. 9, pp. 21–25, 2005.
[13] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *advances in neural information processing systems 15*. cambridge, ma: mit press, 2003, pp. 1547–1554.
[14] S. Schaal, A. Ijspeert, and A. Billard, *Computational approaches to motor learning by imitation.* oxford university press, 2004, no. 1431, pp. 199–218.
[15] P. Andry, P. Gaussier, and J. N. and. B.Hirsbrunner, "Learning invariant sensory-motor behaviors: A developmental approach of imitation mechanisms," *Adaptive behavior*, vol. 12, no. 2, 2004.
[16] T. Chaminade, E. Oztop, G. Cheng, and M. Kawato, "From self-observation to imitation: Visuomotor association on a robotic hand," *Brain Res Bull.*, vol. 6, no. 75, pp. 775–84, 2008.
[17] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensorymotor coordination to imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.
[18] M. Lagarde, P. Andry, and P. Gaussier, "The role of internal oscillators for the one-shot learning of complex temporal sequences," in *Artificial Neural Networks – ICANN 2007*, ser. LNCS, J. M. de Sa, L. A. Alexandre, W. Duch, and D. Mandic, Eds., vol. 4668. Springer, 2007, pp. 934–943. [Online]. Available: http://publi-etis.ensea.fr/2007/LAG07
[19] S. Calinon, F. Guenter, and A. Billard, "On Learning, Representing and Generalizing a Task in a Humanoid Robot," *ieee transactions on systems, man and cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, vol. 37, no. 2, pp. 286–298, 2007.
[20] Y. Demiris and M. Johnson, "Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning," *Connect. Sci.*, vol. 15, no. 4, pp. 231–243, 2003.
[21] V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti, "Action recognition in the premotor cortex," *Brain*, vol. 119, pp. 593–609, 1996.
[22] Y. Demiris, "Imitation, Mirror Neurons, and the Learning of Movement Sequences," in *the International Conference on Neural Information Processing (ICONIP-2002)*, Singapore, 2002, pp. 111–115.
[23] A. N. Meltzoff and M. K. Moore, "Imitation of facial and manual gestures by human neonates," *Science*, vol. 198, no. 4312, pp. 75–78, 1977.
[24] S. S. Jones, "The development of imitation in infancy," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1528, pp. 2325–2335, Aug. 2009.
[25] P. Gaussier, S. Moga, J. P. Banquet, and M. Quoy, "From Perception-Action loops to imitation processes: A bottom-up approach of learning by imitation," *Applied Artificial Intelligence*, vol. 1, no. 7, pp. 701–727, 1998.
[26] G. Metta, G. Sandini, L. Natale, L. Craighero, and L. Fadiga, "Understanding mirror neurons: A bio-robotic approach," *Interaction Studies*, vol. 7, no. 2, pp. 197–232, 2006.
[27] C. Heyes, "Where do mirror neurons come from?" *Neuroscience & Biobehavioral Reviews*, vol. 34, no. 4, pp. 575–583, Mar. 2010.
[28] G. Rizzolatti and L. Craighero, "The Mirror-Neuron System," *Annual Review of Neuroscience*, vol. 27, no. 1, pp. 169–192, 2004.
[29] E. Oztop, M. Kawato, and M. Arbib, "Mirror neurons and imitation: a computationally guided review." *Neural networks : the official journal of the International Neural Network Society*, vol. 19, no. 3, pp. 254–271, Apr. 2006.
[30] P. Gaussier and S. Zrehen, "Perac: A neural architecture to control artificial animals," *Robotics and Autonomous Systems*, vol. 16, no. 2-4, pp. 291–320, December 1995.
[31] C. Giovannangeli and P. Gaussier, "Interactive Teaching for Vision-Based Mobile Robots: A Sensory-Motor Approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 40, no. 1, pp. 13–28, 2010.
[32] S. Schaal, "Dynamic movement primitives - a framework for motor control in humans and humanoid robots," in *the international symposium on adaptive motion of animals and machines*, 2003.
[33] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *international conference on robotics and automation (icra2009)*, 2009.
[34] S. Schaal and C. G. Atkeson, "Constructive Incremental Learning from Only Local Information," *Neural Comput.*, vol. 10, no. 8, pp. 2047–2084, 1998.
[35] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," *Robotics and Autonomous System*, vol. 16, no. 2-4, pp. 333–356, December 1995.
[36] ——, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2007, pp. 255–262.
[37] E. L. Sauser, B. D. Argall, G. Metta, and A. G. Billard, "Iterative Learning of Grasp Adaptation through Human Corrections," *Robotics and Automation Systems*, pp. 1–17, 2011.
[38] S. Calinon, F. D'halluin, D. G. Caldwell, and A. Billard, "Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework," in *Proceedings of 2009 IEEE International Conference on Humanoid Robots*, 2009, pp. 582–588.
[39] C. Giovannangeli, P. Gaussier, and G. Désilles, "Robust Mapless Outdoor Vision-based Navigation," in *IEEE/RSJ International Conference on Intelligent Robots and systems*. Beijing, China: IEEE, 2006.
[40] M. Maillard, O. Gapenne, L. Hafemeister, and P. Gaussier, "Perception as a dynamical sensori-motor attraction basin," in *Advances in Artificial Life, 8th European Conference, ECAL 2005, Canterbury, UK, September 5-9, 2005, Proceedings*, ser. Lecture Notes in Computer Science, M. S. Capcarrre, A. A. Freitas, P. J. Bentley, C. G. Johnson, and J. Timmis, Eds., vol. 3630. Springer, 2005, pp. 37–46.
[41] N. Hogan, "Impedance Control: An Approach to Manipulation," in *American Control Conference, 1984*, Department of Mechanical Engineering, Laboratory of Manufacturing and Productivity, Massachusetts Institute of Technology, Cambridge, MA 02139. IEEE, Jun. 1984, Conference proceedings (article), pp. 304–313.
[42] G. A. Carpenter and S. Grossberg, "Adaptive resonance theory (ART)," in *The handbook of brain theory and neural networks*. Cambridge, MA, USA: MIT Press, 2002, pp. 79–82.
[43] G. SCHONER, "Dynamics of behavior: Theory and applications for autonomous robot architectures," *Robotics and Autonomous Systems*, vol. 16, no. 2-4, pp. 213–245, Dec. 1995.
[44] P. Andry, P. Gaussier, J. Nadel, and B. Hirsbrunner, "Learning Invariant Sensorimotor Behaviors: A Developmental Approach to Imitation Mechanisms," *Adaptive Behavior*, vol. 12, no. 2, pp. 117–140, Jun. 2004.
[45] S.-I. Amari, "Dynamics of pattern formation in lateral-inhibition type neural fields," *Biological Cybernetics*, vol. 27, no. 2, pp. 77–87, 1977.
[46] A. Billard and M. J. Matarić, "Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture," *Robotics and Autonomous Systems*, vol. 37, no. 2-3, pp. 145–160, 2001.
[47] C. L. Nehaniv and K. Dautenhahn, "The correspondence problem." MIT Press, 2002, pp. 41–61.
[48] JP.Banquet, P.Gaussier, JC.Dreher, C.Joulain, A.Revel, and W.Gunther, *Space-time, order and hierarchy in Fronto-hippocampal system: a neural basis of personality*. Elsevier Science, Amsterdam, 1997, ch. 4, pp. 123–189.
[49] M. Lagarde, P. Andry, P. Gaussier, S. Boucenna, and L. Hafemeister, "Proprioception and Imitation: On the Road to Agent Individuation," in *From Motor Learning to Interaction Learning in Robots*, O. Sigaud and J. Peters, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 264, book part (with own title) 3, pp. 43–63.

[50] J. Hirel, P. Gaussier, and M. Quoy, "Model of the Hippocampal Learning of Spatio-temporal Sequences," in *Artificial Neural Networks – ICANN 2010*, vol. 6354, 2010, pp. 345–351.

[51] M. Lagarde, P. Andry, P. Gaussier, and C. Giovannangeli, "Learning new behaviors : Toward a control architecture merging spatial and temporal modalities," in *Workshop on Interactive Robot Learning - International Conference on Robotics: Science and Systems (RSS 2008)*, June 2008.

[52] J. Hirel, P. Gaussier, and M. Quoy, "Biologically inspired neural networks for spatio-temporal planning in robotic navigation tasks," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, 2011, pp. 1627–1632.

[53] A. de Rengervé, S. Boucenna, P. Andry, and P. Gaussier, "Emergent Imitative Behavior on a Robotic Arm Based on Visuo-Motor Associative Memories," in *IEEE/RSJ International Conference on Intelligent Robots and systems (IROS'10)*, Taipei, Taiwan, 2010, pp. 1754–1759.

[54] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, 2009.

[55] S. Calinon and A. Billard, *Learning of Gestures by Imitation in a Humanoid Robot*, k. dautenhahn and c.l. nehaniv ed. Cambridge University Press, 2006, in press.

[56] S. Schaal, "Learning from demonstration," in *advances in neural information processing systems 9*. mit press, 1997, pp. 1040–1046.

[57] P. Zukowgoldring and M. Arbib, "Affordances, effectivities, and assisted imitation: Caregivers and the directing of attention," *Neurocomputing*, vol. 70, no. 13-15, pp. 2181–2193, Aug. 2007.

[58] J. P. Banquet, P. Gaussier, M. Quoy, A. Revel, and Y. Burnod, "Cortico-hippocampal maps and navigation strategies in robots and rodents," in *Proceedings of the seventh international conference on simulation of adaptive behavior on From animals to animats*, ser. ICSAB. Cambridge, MA, USA: MIT Press, 2002, pp. 141–150.

[59] N. Cuperlier, M. Quoy, and P. Gaussier, "Neurobiologically inspired mobile robot navigation and planning." *Front Neurorobotics*, vol. 1, p. 3, 2007.

[60] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 1998.

[61] P. Redgrave, T. J. Prescott, and K. Gurney, "The basal ganglia: a vertebrate solution to the selection problem?" *Neuroscience*, vol. 89, no. 4, pp. 1009–1023, 1999.

[62] A. de Rengerve, J. Hirel, P. Andry, M. Quoy, and P. Gaussier, "On-line learning and planning in a pick-and-place task demonstrated through body manipulation," in *IEEE International Conference on Development and Learning (ICDL) and on Epigenetic Robotics (Epirob), 2011*, vol. 2. Frankfurt am Main, Germany: IEEE, Aug. 2011, pp. 1–6.

[63] T. Nielsen, "Volition: A new experimental approach," *Scandinavian Journal of Psychology*, vol. 4, pp. 225–230, 1963.

[64] M. Jeannerod, "To act or not to act. perspectives on the representation of actions," *Quaterly Journal of Experimental Psychology*, vol. 52A, pp. 1–29, 1999.

[65] J. N. et C. Potier, "Imitez, imitez, il en restera toujours quelque chose: le statut developpemental de l'imitation dans le cas d'autisme," *ENFANCE PARIS*, pp. 76–85, 2002.

[66] P. Andry, P. Gaussier, S. Moga, J. Banquet, and J. Nadel, "Learning and communication in imitation: An autonomous robot perspective," *IEEE transactions on Systems, Man and Cybernetics, Part A*, vol. 31, no. 5, pp. 431–444, 2001.

[67] A. Revel and P. Andry, "Emergence of structured interactions: From a theoretical model to pragmatic robotics," *Neural Network*, vol. 22, pp. 116–125, 2009.

[68] P. Andry, A. Blanchard, and P. Gaussier, "Using the rhythm of nonverbal human-robot interaction as a signal for learning," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 3, 2011.

[69] S. Vijayakumar, A. D'souza, and S. Schaal, "Incremental Online Learning in High Dimensions," *Neural Comput.*, vol. 17, no. 12, pp. 2602–2634, Dec. 2005.

[70] I. Fukuyori, Y. Nakamura, Y. Matsumoto, and H. Ishiguro, "Flexible Control Mechanism for Multi-DOF Robotic Arm Based on Biological Fluctuation," *From Animals to Animats 10*, pp. 22–31, 2008.

[71] G. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956.

[72] F. Gobet, P. C. R. Lane, S. Croker, P. C. H. Cheng, G. Jones, I. Oliver, and J. M. Pine, "Chunking mechanisms in human learning," *Trends in Cognitive Sciences*, vol. 5, no. 6, Jun. 2001.

[73] S. Boucenna, P. Gaussier, L. Hafemeister, and K. Bard, "Autonomous development of social referencing skills," in *From Animals to Animats 11*, 2010, vol. 6226, pp. 628–638.

[74] A. de Rengerve, R. Braud, P. Andry, and P. Gaussier, "Behavior adaptation from negative social signal based on goal awareness," in *2012 IEEE International Conference on Development and Learning (ICDL) - Epigenetics and Robotics (Epirob)*, San Diego, CA, USA, Nov. 2012, pp. 1–6.

[75] S. Boucenna, P. Gaussier, and L. Hafemeister, "Development of joint attention and social referencing," in *2011 IEEE International Conference on Development and Learning (ICDL)*, vol. 2. IEEE, Aug. 2011, pp. 1–6.

[76] S. Klanke, S. Vijayakumar, and S. Schaal, "A Library for Locally Weighted Projection Regression," *Journal of Machine Learning Research*, vol. 9, pp. 623–626, 2007.

[77] B. Widrow and M. E. Hoff, "Adaptive Switching Circuits," in *1960 {IRE} {WESCON} Convention Record, Part 4*. New York: IRE, 1960, pp. 96–104.

PLACE PHOTO HERE

Antoine Biography text here.

Pierre Biography text here.

Philippe Biography text here.

3.2 Exploiter les signaux de renforcement négatifs pour modifier les buts utilisés dans une tâche

La compréhension des intentions des autres permet d'anticiper les actions et d'améliorer la coopération entre deux individus [Tomasello et al., 2005]. La théorie de la simulation motrice [Gallese, 1998; Iacoboni et al., 2005] explique cette reconnaissance des intentions de l'autre en s'appuyant sur les propriétés du système des neurones miroirs [Gallese et al., 1996; Rizzolatti and Craighero, 2004] pour reconnaître les actions observées (Sec. 1.5.2). On peut distinguer la reconnaissance des intentions motrices (les buts) et la reconnaissance des intentions préalables (pouvant être liées à un aspect plus social) [Jacob and Jeannerod, 2005]. Dans un contexte d'interaction sociale entre un robot et un humain, on peut aussi s'intéresser au fait que l'humain pourra interpréter les intentions du robot. Les renforcements fournis par l'humain peuvent porter aussi bien sur l'action directement perçue que sur l'intention supposée de l'humain. Cette constatation est particulièrement vraie dans le cas d'un retour négatif donné par un professeur humain. [Thomaz and Breazeal, 2007] se sont intéressés à l'usage des retours négatifs par des sujets naïfs dans un apprentissage par renforcement. Ils soulignent le fait que si les retours positifs viennent plutôt récompenser le passé, les retours négatifs portent sur l'action courante, que nous interprétons comme une remise en cause du but suivi par le robot. Dans le cas où le robot se trompe de but, le retour négatif est généralement donné dès que le professeur humain perçoit que l'intention motrice du robot est incorrecte. Le professeur n'attendra donc pas que le robot termine la fin de la séquence erronée pour lui signifier son erreur. Le robot doit modifier rapidement son comportement en conséquence. Cela implique que le robot doit adapter l'association entre le contexte et le but sur la base du signal donné en anticipation (puisque le robot n'a pas encore atteint son but). Nous illustrons cette situation par l'exemple de la figure 3.1. Un contexte motivationnel représentant la soif d'un agent naviguant peut être associé avec 2 buts (sources d'eau notées Goal1 et Goal2 sur la figure). Selon l'activation du contexte et la position initiale du robot, celui-ci choisira d'aller à l'un ou l'autre des deux buts. Dans le cas de la figure 3.1, le robot se dirigera vers la position Goal2 car elle est la plus proche. Un professeur humain peut décider que le but Goal2 est interdit, ce qui se traduira par des signaux de renforcements négatifs si le robot continue de s'en approcher. La première réaction du robot pourrait être d'inhiber l'action courante au moment où le signal négatif est perçu. Cependant dans ce cas, le robot continuera d'avancer vers Goal2 car il s'agit toujours de la source la plus proche malgré le détour induit par la contrainte de ne pas prendre un chemin directe (Fig. 3.1a). Pour éviter que le comportement erroné persiste, le but correspondant à la source interdite devrait être inhibé voire désassocié complètement du contexte actif. Le robot pourra alors poursuivre un autre objectif et corriger ainsi son comportement en allant vers Goal1 (Fig. 3.1b).

Dans l'article de la section suivante, nous présenterons comment adapter notre modèle de carte cognitive pour découvrir quel but inhiber. Chaque but actif propage une activité à travers la carte cognitive avec les potentiels diffusés entrant en compétition à chaque étape de la propagation. Le comportement du robot sera donc décidé par le but dont le potentiel donne la valeur du gradient perçu dans l'état courant. Comme le gradient est propagé de la même manière quelque soit le but, le robot ne peut pas savoir directement quel est le but poursuivi. Afin de déterminer le but suivi, nous utiliserons le fait que, dans une carte cognitive, seul une variation sur le but poursuivi pourra générer une variation du potentiel lu pour l'action sélectionnée. Suivant cette constatation, il suffira de modifier le potentiel d'un but et d'observer le résultat produit sur le

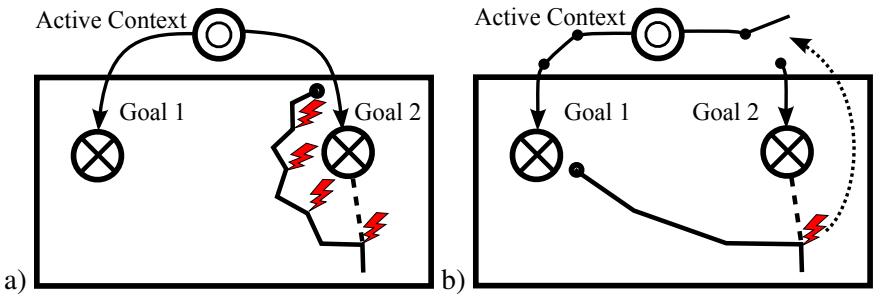


FIGURE 3.1 – Modification du comportement d'un agent sur la base d'un retour négatif. L'agent motivé navigue dans un environnement avec 2 ressources positionnées en Goal1 et Goal2. Le besoin actif motive le robot à rejoindre l'un des deux emplacements de ressources. En parallèle, un agent "éclairé" peut donner un retour négatif non verbal à l'agent motivé afin de l'empêcher de rejoindre l'emplacement noté Goal2 (trajectoire en pointillé). *Gauche* Le signal négatif déclenche l'inhibition locale de la direction de mouvement courante. Comme l'agent continue de considérer que Goal2 est une destination valide, plusieurs signaux négatifs doivent être fournis pour parvenir à changer l'objectif de l'agent. *Droite* Le signal négatif retire directement l'attraction liée à Goal2. L'agent motivé redirige immédiatement sa trajectoire vers Goal1.

potentiel de l'action sélectionnée. On pourra ainsi trouver le but qui guide le comportement du robot. La connaissance du but suivi permettra, lorsqu'un renforcement négatif est perçu, de désapprendre l'association entre le contexte actif et ce but. Si un autre but l'emporte alors, le comportement aura bien été modifié. Nous montrerons les résultats obtenus en simulation dans une expérience de navigation entre différents buts et nous validerons l'utilisation de l'information du but suivi dans le cas où un professeur humain donne un renforcement négatif pour interdire un but.

3.2.1 Article3 : Reconnaissance explicite des buts suivis et inhibition d'intention motrice suivant des retours négatifs dans une expérience de navigation simulée

de Rengerve, A., Braud, R., Andry, P., and Gaussier, P. (2012). Behavior Adaptation from Negative Social Signal Based on Goal Awareness. In *2012 IEEE International Conference on Development and Learning (ICDL) - Epigenetics and Robotics (Epirob)*, pages 1–6, San Diego, CA, USA

Behavior adaptation from negative social signal based on own goal awareness

Antoine de Rengervé, Raphael Braud, Pierre Andry and Philippe Gaussier
ETIS, CNRS ENSEA University of Cergy-Pontoise, F-95000 Cergy-Pontoise, France
{rengerve,raphael.braud,andry,gaussier}@ensea.fr

***Abstract*—Robots are expected to perform actions in a human environment where they will have to learn both how and when to act. Social human robot interaction could provide the robot with external feedback to guide them. In previous work, we have developed bio-inspired models for action planning which enables the system to adapt its representations and thus its behavior in the context of latent learning with rewards. In this paper the focus is put on using negative signals. It stresses an important feature of a cognitive system : it must be aware of its own objectives i.e. aware of what it is about to do. The model presented here allows the robot the awareness of its goal, and we show that such a knowledge enhances the behavior of a robot receiving an external negative signal.**

I. INTRODUCTION

Robots are expected to enter in a closer and closer interaction with humans. They should be able to act on the world accordingly. Working in a human environment requires that robots can adapt to a changing environment i.e. with constraints on when and how to perform actions that can evolve. During human robot interaction, the robot is expected to follow human instructions. In pre-verbal stage, the feedback modulating the actions of the robot could be a simple social feedback like facial expressions. Social referencing [1] corresponds to the observed fact that infants can use their parents' expression to valuate an object, a situation or an action. Social referencing was implemented on robots [2] [3] using this social feedback to determine whether or not they could play with a given object. If an expression of joy is presented, the robot knows that it can reach for the object whereas an expression of anger or fear will make the robot avoid touching the object. The same feedback can also be used to modulate directly behaviors for instance by weighting some sensorimotor associations in navigation [4].

Let us consider the case of an agent planning actions in its environment. Given a certain context, its behavior may unfold as different actions and specific goals that would terminate these sequences of actions. Basically, reaching the correct goal can give the agent a reward and usually changes the active context. For instance, the behavior could be navigating and getting resources like water at different places (goals) when the agent is thirsty. As several water resources may be available in the environment, the robot could reach any of them to satisfy it thirst. There is also a knowing agent (like a human) that can help the robot to decide where to go. The knowing agent can convey a negative signal when it sees that the robot is making wrong choices of action (e.g. going to a dried-out well). A negative feedback is usually given as soon as the human teacher notices that the robot is doing something wrong i.e. it will not wait until the end of the sequence to show the robot that it is making mistakes. The mistake may be the action and thus the performed action may be inhibited. However, if it is the goal that was incorrect, only inhibiting actions is not efficient to change the behavior of an agent that tries to reach a wrong goal (Fig. 1a). If the agent had access to the information of the pursued goal, it would be able to remove the activation of the incorrect goal. Then the agent would pursue another goal changing adequately its behavior (Fig. 1b). How can the robot update a context-goal association from an anticipatory given signal ? To do so, the robot will have to be aware of its goals and motivations in order to change them. Such a knowledge could help it to solve this immediate planning problems of inhibiting the target of a behavior, but it would more generally be useful for the agent to have a better control over its own behaviors.

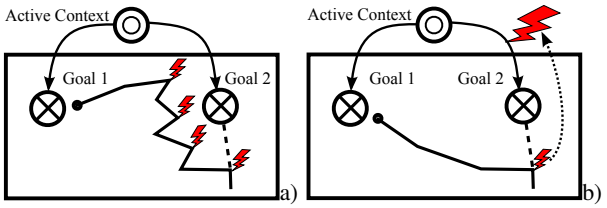


Fig. 1. Modifying an agent behavior from negative signals. A motivated agent navigates in an environment with two resources placed at Goal1 and Goal2. The motivational context (drive) for reaching one of the resources is active. A knowing agent conveys negative signals to prevent the motivated agent from going to Goal2 (trajectory in dash line). *Left* The negative signals trigger the inhibition of the directions of movement. As the agent still considers the Goal2 as a correct goal, many negative signals are needed to change the target of the agent. *Right* The negative signal removes the attractiveness of the second goal. The agent now aims directly for Goal1.

In previous works, we developed models explaining how a robot can navigate and even plan its navigation. These models use place-cells, a particular type of neuron found in the Hippocampus that maximally fires when the robot is at a learned spatial position. At first, these place-cells can be directly associated with orientations i.e. direction to be heading to. Such simple sensorimotor associations can build attractors defining trajectories in the space [5] [6]. However, with such a model, action planning is limited to using reinforcement learning [7] that can be quite long to adapt to changes. A latent

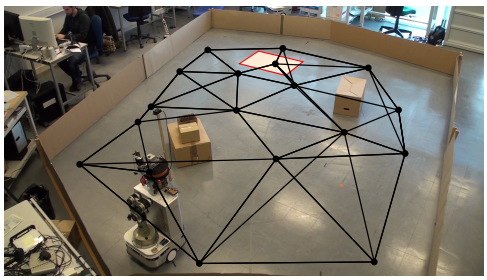


Fig. 2. Cognitive map built on-line during visual navigation task with Robulab (Robosoft) mobile platform.

learning of a topological model of the environment can build faster representations of the possible actions and can adapt plans faster. Those topological maps, called cognitive maps, are based on encoded actions that are transitions between place-cells associated with orientations of movement [8]. An example of built cognitive map is given in Fig. 2. Cognitive maps can encode the possible sequences of transitions between place-cells as recurrent connections. A drive corresponding to a motivational context or an active physiological need (e.g. thirst) can be associated with one of the transitions in the cognitive map. With respect to the motivation, this transition then represents a goal for the system. As a result of the recurrent network encoding, a gradient of activities is diffused from the goal to the other transitions in the cognitive map. These propagated activities can give a bias on the transitions to be performed. The selected direction of the movement then enables the navigating robot to follow the shortest path toward its goal [9] [8]. The use of cognitive maps are not restricted to transitions between place-cells and navigational task. The categories encoded and used in the Hippocampus may correspond to multi-modal states [10]. This idea was implemented on real robots with a cognitive map based on proprioceptive states and color based motivational contexts that enabled a robot made of a robotic arm and a camera to sort colored cans [11]. Whatever the task is, the action planning with cognitive maps relies on a gradient ascent on the diffused activities. The selected actions lead the agent to the closest goal that is a local maximum of the gradient. The robot cannot know where is the local maximum before it is reached. As it does not have a direct access to the goal it is pursuing, how can an agent determine from the diffused gradient what its current goal is ?

In Section II, the model of cognitive maps that is used in this paper is briefly summarized. A specific focus is given on how the goals are encoded in the described implementation of the cognitive maps. In Section III, we detail how an agent that chooses actions on the basis of a gradient ascent can determine its own objectives. The mechanisms are first to select and inhibit one of the possible goals and then to monitor if it is related to the current behavior of the agent. In that case, the agent succeeded to estimate its goal and the result is the selected goal. In Section IV, a simulated agent goes through the different steps to built the representations for planning with motivational contexts. The goal awareness system is implemented and enables the robot to modify its behavior when an external negative signal is perceived. In Section V, we remind the biological relevancy of this model of cognitive map and we discuss its position in the development of planning capabilities.

II. Cognitive map and goal pursuit

The cognitive map model relies on the computation of the performed transitions between different states. In the case of navigation, each state corresponds to a place-cell that fires maximally when the robot is at the location which is encoded by the cell. The predicted transitions are used to build the cognitive map. Each time a transition is performed, the cognitive map is updated to include this transition into the topological graph of the possible sequences of transitions. In the cognitive map, recurrent connections between neurons representing the different transitions are adapted as the robot behaves. The activity from recurrent network o_j^{rec} is the result of the competition between the different activities propagated through the recurrent connections.

The system is considered to be in an exclusive motivational state m given by the drive layer, also called the motivational context layer. In this layer, only one neuron (index m) can be different from null and equal to 1. In previous works [12], the motivational contexts were directly associated with some neurons in the cognitive map implicitly defining the corresponding transition as a goal. In order to manipulate the goals more easily they are now recruited in a separate layer. The recruitment of a new learned goal L is done when a reward is received (R=1) (eq. (1)). A goal is directly related to the last performed transition L assumed to be the one that get the reward. The learning is based on a recruitment according to a vigilance threshold and a Hebbian like rule for the maximally activated neurone L_J. The learning depends on a learning rate and a decay factor, α^L, supposed equal to enable the convergence of the weights toward 1.

$$\begin{cases} L_j = \sum_i w_{ij}^L \cdot T_i^L \\ \Delta w_{iJ}^L = R \cdot \alpha^L (T_i^L \cdot L_J - w_{iJ}^L) \end{cases} \quad (1)$$

The learned goals L are gated by the reception of a reward. Only when a reward is received the correlation between an active drive M_m and an active learned goal will be learned with the following Hebbian like rule (2). The resulting activities in the learning layer corresponds to the desire of performing this goal, called a desired goal D, given an active drive.

$$\begin{cases} D_j = [\sum_i w_{ij}^D \cdot M_i + 2R \cdot \mathcal{H}(L_j) - R]^+ \\ \Delta w_{ij}^D = \alpha_j^D (M_i \cdot D_j - \cdot w_{ij}^D) - \lambda_j^D \cdot w_{ij}^D \cdot M_i \\ \text{with } \alpha_j^D = R \cdot \varepsilon^D \cdot \mathcal{H}(L_j) \end{cases} \quad (2)$$

where $\mathcal{H}$ is the Heavyside function and ε^D is the global learning rate and with a topological neuromodulation α_j^D of the learning given by the learned goals L and gated by the

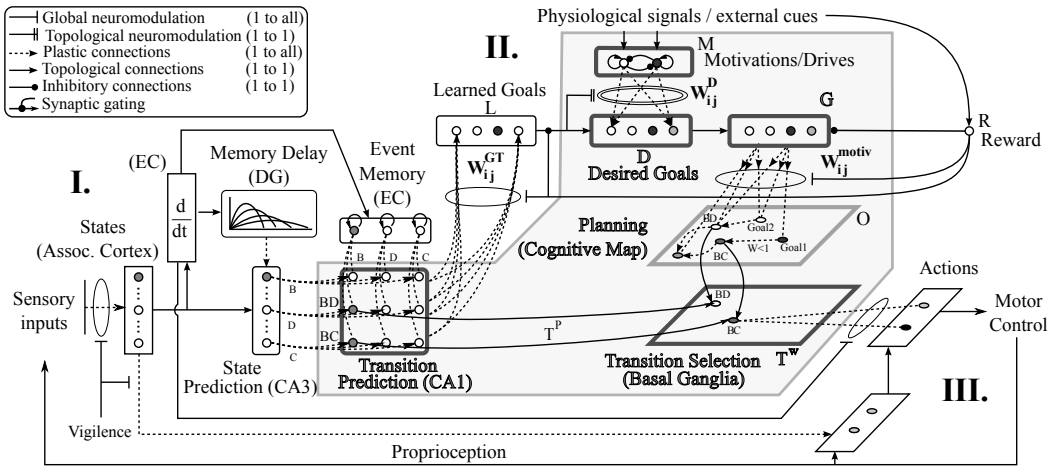


Fig. 3. Cognitive map based motor control and own goal estimation. I.) States representing the places are recruited with respect to a vigilance threshold. Changes of states are events that are predicted from memory delay memorizing the timing of the change. Based on these events, transitions are predicted. II.) When a reward is received, the last performed transition is encoded as a goal. The reward and the active goal supervise the association between the active drive with the goals. Desired goals encode the confidence in getting a reward related to the active drive. The desired goal layer projects these activities in the cognitive map. The cognitive map also learns the possible sequences of transitions. A gradient activity propagates from the active goal-transition to the previous transitions in the learned sequences. III.) The cognitive map activities can bias the selection of the transition to be performed. Doing so the system follows the gradient toward the topologically closest goal.

reward R. The active decay λ_j^D can be used to unlearn the drive-goal association. It is always null except when a negative feedback is received (see eq. (14)). It must be noted that during reward reception there will only be one active desired goal. Thus, the reward only reinforces the association between the active drive and the goal corresponding to the current last performed transition (index k in the cognitive map). An intermediary layer G, receiving exciting connection from the desired goal layer D, must be introduced here. Currently it is only a copy of the activity in D. Its role appears clearly during the goal selection and inhibition during the goal detection process described in Section III. The connections from the goal layer G to the cognitive map are also learned (3).

$$\Delta w_{ij}^{motiv} = R(\delta_{jk} \cdot \varepsilon^{motiv} \cdot D_i - \alpha^{motiv} \cdot w_{ij}^{motiv}) \quad (3)$$

In the cognitive map, only the neuron corresponding to the last performed transition (index k) is active. As the recruitment ensure that only one goal is associated to a given transition, a transition in the cognitive map can only be associated with this unique learned goal. A motivational context inputs an activity o_j^{motiv} in the cognitive map (4), considering the goals desired after inhibition G.

$$o_j^{motiv} = [\max_i(w_{ij}^{motiv} \cdot G_i)]^+ \quad (4)$$

The output activities O_j of the neurons in the cognitive map result from a competition between the computed activities from the recurrent connections o_j^{rec} and the activities o_j^{motiv} related to the motivations (drives) of the agent (5)[1].

$$O_j = [max(o_j^{motiv}, o_j^{rec})]^+ \quad (5)$$

[1]In the following description of the model (Fig. 4), the equations are given for discrete time.

The motivational activities are diffused from the associated transitions to the previous transitions and so on with decreasing activities as the recurrent connective weights are lower than 1. The activities in the cognitive map come to bias the selection of the transition T_b^W that determines the motor commands (eq. 6).

$$\begin{cases} T_j^W = \delta_{jb} \cdot \mathcal{H}(T_j^s) \\ b = \underset{j}{argmax}(T_j^s)) \\ T_j^s = max((T_j^P - 1) + O_j) \end{cases} \quad (6)$$

with T^P the possible transitions in a given state. As a result of the different competitions, the activity of each neuron in the cognitive map corresponds to only one gradient, resulting of the activity of the different goals.

III. Detecting own objectives from gradient propagation in a learned cognitive map

The principle of the goal detection is to modify the diffused gradients by modifying goals activities, one after another, and to monitor if the modifications are propagated to the activity of the selected transition T^W (Figure 4). The desired goal activities can be modulated by the research of the current followed goal. Goals are successively selected and tested.

An internally built "keep goal" signal K supervises the goal checking by gating the selection of a new goal. When it is null, a new goal can be selected to modulate the desired goal activities. Otherwise the K signal is equal to 1, and it maintains the selected goal until the checking processed is finished or as long as required to keep the result when the detection is successful. The goal checking process depends on the propagation of modifications of the gradients in the cognitive map. Once the running propagation signal P (eq. (7))

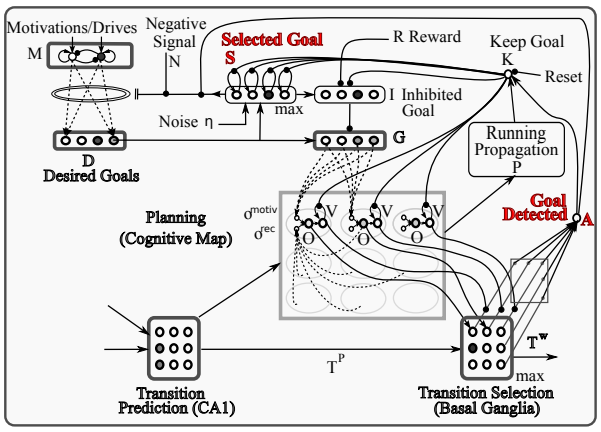


Fig. 4. The model of cognitive map is extended to let the agent be aware its goals. The thick lines blocks of the gray area in Figure 3 are displayed with the added blocks. One of the desired goals is selected to briefly inhibit one of the goal activities G to be input in the cognitive map. The wave of modulation of activities propagates in the cognitive map to any neurons representing a transition related to the selected goal. If the variation monitoring V matches the current selected transition T^W, the selected goal effectively determines to the current behavior. If the activities in the cognitive map converge (no more modulation propagation) without successful detection, then another goal is selected to be checked. When the goal is detected, the result is in the selected goal layer. When a negative signal is received, the connection between the active drive and this selected goal is decayed reducing the activity of this goal in the desired goals layer.

stops detecting changes in the cognitive map, the K signal can become null again unless the goal detection is successful ($A = 1$) (eq. (8)).

$$P = \sum_j \mathcal{H}(O_j(t-1) - O_j(t)) + \mathcal{H}(O_j(t) - O_j(t-1)) \quad (7)$$

$$K = \mathcal{H}(A(t-1) + P(t-1) - reset) \quad (8)$$

A reset signal can also be applied to force a new a goal detection[2].

The activities in the selected goal layer S is modified only when K is null. A new goal (index a) is selected depending on the current possible goals estimated from the desired drives D (binary values) and some noise η (eq. 9).

$$\begin{cases} S_j = & \delta_{ja} \cdot \mathcal{H}(s_j) \\ a = & \underset{j}{argmax}(s_j) \\ s_j = & [\, K \cdot S_j(t-1) \\ & +(1-K)(\, (\mathcal{H}(D_j) - 1) \\ & \quad +A(t-1) \cdot S_j(t-1) + \eta \,)\,]^+ \end{cases} \quad (9)$$

In order to perform the goal detection, the selected goal are inhibited one after the other. The inhibition goal layer I receives the selected goal S activities and some inhibition from the reward R and the keep goal signals K (eq. (10)). The reward signal R can prevent the modulation in order to avoid perturbing the learning of the goal to cognitive map associations.

$$I_j = \mathcal{H}(S_j - K - R) \quad (10)$$

The goal layer G contains the desired goal D activities modulated by the goal inhibition I.

$$G_j = D_j - 0.5 I_j \quad (11)$$

The success of the detection is stored in the goal detected signal A meaning the detection have been achieved (eq. (13)). From (9), a selected goal a is detected as the current goal if it generates the propagated gradient that gives the activity of the current transition to be performed. Neurons in the layer V are dedicated to detecting strong negative variations of the propagated activities in the cognitive map. The layer keeps the activations in memory as long as no new goal is checked (12). The goal-detected signal is activated only if one of the active neurons in the variation detection layer corresponds to the selected transition in T^W.

$$V_j = \mathcal{H}(O_j(t-1) - O_j(t)) + V_j(t-1) \cdot K \quad (12)$$

$$A = \mathcal{H}(\sum_j V_j \cdot T^W_j) \quad (13)$$

Thereby, the own goal evaluation is based on simple mechanisms: selecting a goal, modulating its propagation and monitoring if it influences the propagated activity at the level of the selected transition. The information of which goal is pursued is important to let the agent have a better control over its own behavior. For instance, the information of the current goal can be used to reduce the desire for this goal when a negative signal is received. In the equation of the drive-goal association learning (eq. (2)), a topological active decay term λ^A_j is introduced. This term can be modulated to ensure that the association is unlearned when a negative feedback is received (eq. 14).

$$\lambda^A_j = \lambda^A \cdot N \cdot A \cdot S_j \text{ with } \lambda^A = 0.5 \quad (14)$$

with λ^A a global decay factor. If a negative feedback N is received while the goal detection is successful ($A = 1$), then the detected goal present in S will enable the decay of the connection between this goal and the active drive. As the necessary signals are already present, the mechanism to inhibit the behavior is then very simple.

IV. Behavior Inhibition in an Agent Aware of its Goals

The model for goal awareness was tested in a simulation of an autonomous agent navigating in a Cartesian 2D space. The basis of the simulation is a quite classic paradigm of autonomous motivated navigation. The agent is to build the corresponding action representations. Then, some interactions with the agents will be used to modify the behavior of the robot with the use of the goal awareness system. The virtual agent needs are food and water. In the environment, two water

[2]Each signal is noted without the time index when it correspond to the current iteration. The time index is only given when it differs from the current iteration like $(t-1)$ for previous iteration.

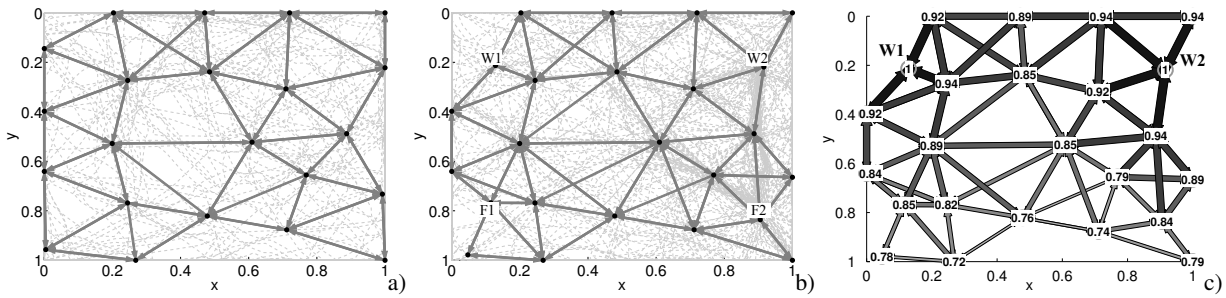


Fig. 5. *Left* The agent explore its environment and encoded it as states (place-cells), transitions and sequences of transitions. Dots are place-cells, arrows are learned oriented transitions and the dash-gray lines are the trajectories of the robot during exploration. *Center* Infinite resources (water: $W1, W2$ and food: $F1, F2$) are added to the environment. The agent continue to explore and learn the rewarded goals. *Right* Resulting cognitive map built during these two first phases. The arrows and their thicknesses and colors represent respectively the possible transitions and their activities in the cognitive map (the bigger and darker for higher values). The activities correspond to the propagated gradient when the first drive (thirst) is active.

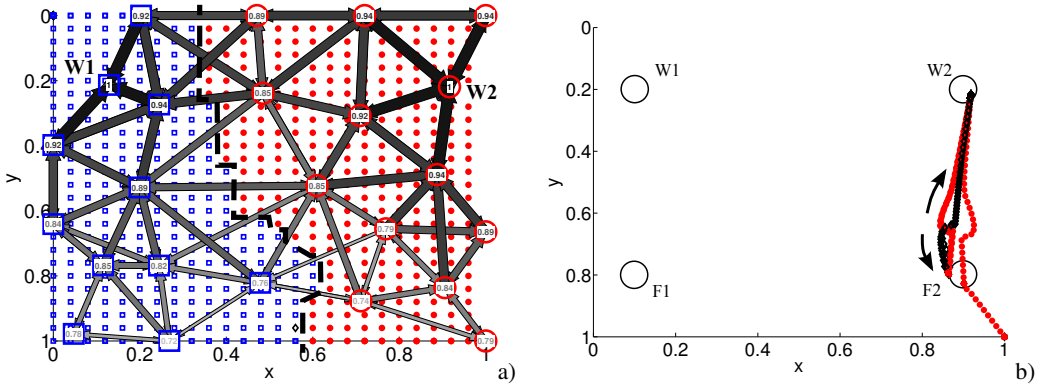


Fig. 6. *Left* Representation of the goal detected by the agent after the cognitive map is built. For each position, the agent estimates the current goal it would pursue. Red dots correspond to the goal on the right (W2) and blue squares are for the goal on the left (W1). The black thick line separates the two areas related to each goal. *Right* Initial behavior of the agent. The agent exploits the two resources on the right (Water 2 and Food 2). The agent always estimates the goal it is going to. The color and the shape of the points of the trajectory correspond to this goal estimation in the case of thirst motivation. Red dots are for W2, blue squares for W1 and black diamond for goals that are not associated with thirst (F1 and F2).

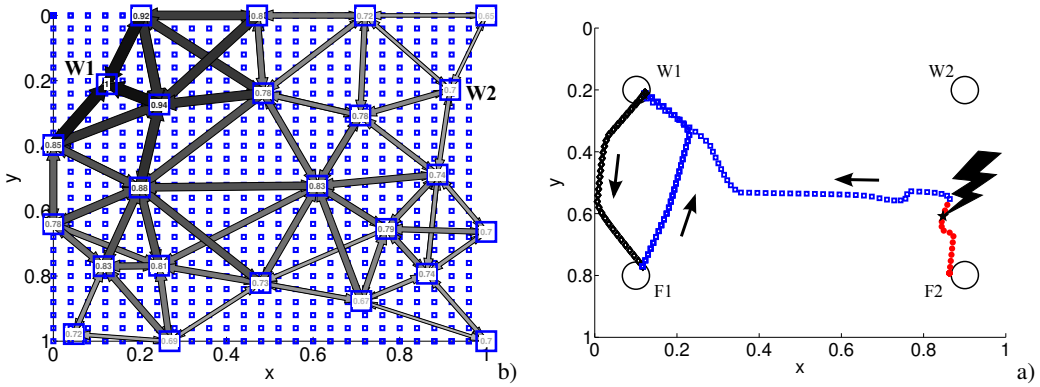


Fig. 7. *Left* Representation of the goals detected by the agent after the negative signal is received. The association between thirst and the goal W2 was reduced by half. Thus, the propagated gradient now only corresponds to the goal W1 (blue squares). *Right* A negative signal is given while the robot moves toward W2. As the propagated gradients changed also does the target of the agent. The change of goal is visible in the goal estimation displayed for each point of the trajectory (blue squares replace red dots). The consecutive behavior of the agent is effectively adapted. It then aims to the goal W1 and keeps exploiting the two goals W1 and F1.

sources and two food sources (all infinite) can be available (see Fig. 5b). As time passes, food and water drives increase. When they are still low, the behavior of the agent is exploratory. When one of them is over a given threshold, a competition selects the most activated drive to motivate the behavior of the agent. Depending on its representation of the environment and the possible actions in it, the agent will reach for the closest adequate resource. When the agent is at the spot, the satisfied drive is reset letting the agent pursue the other drive, if activated enough, or resume its exploratory behavior. In the first phase, the agent explores its environment randomly. Place-cells like categories are recruited when the position of the robot is too different from any already encoded position with respect to a vigilance threshold. In the experiments, a simplification of the visuo-motor based architecture [5] is used. Place-cells are not built on landmark-azimuth couples but on Cartesian position (x, y). Also, the movements of the agent are not directly encoded as an orientation neural field but as the position of the next place-cell to be reached. The goal estimation mechanisms does not depends on the kind of states used but on the goal inhibition and the activity propagation in a topological map. As the agent moves, the maximally recognized category can change. Consecutive categories are then coded as transitions. The possibles successions of transitions are encoded in the cognitive map. The resulting representation of the possible actions in the environment is displayed in Figure 5c.

The capability of the agent to recognize its goals entirely relies on the previous learning and encoding of the cognitive map. After the phase of learning with the resources, the representation given by the own goal detection is evaluated. For different positions in the environment, the agent estimates the goal it expects to reach (Fig. 6). The result stresses the fact that what is important is the state in which the robot is. In each state only one goal is ever expected.

In Fig. 7, a negative feedback is received while the agent is exploiting the resources. It is directly converted into a decay of the drive-goal association reducing it by half. As a result, the diffused gradient is modified and the agent changes its goal and thus its behavior. The resulting goal estimation for each position is given in Fig. 7a. With such a strong decay, the former goal does not propagate anymore because its related motivational activity is lower than the gradient propagated by the other strongly activated goal.

The choice of the action to be performed by the robot is given by both the drive and the state in which the robot is. Depending on its state, some transitions will be possible or not and the propagated gradient may not come from the same goal. The most important effect of the negative feedback is not to reduce the desirability of the goal transition but to modify the area of domination for each goal. In comparison with our previous work [12], goals are now stressed as a major actor of the planning process. They are explicitly coded and can be used to make hypotheses ("Is this one the current goal?") and modulate them in order to find and select the current followed goal.

V. DISCUSSION

In this paper, we studied how negative signals can induce behavior adaptation in the case of action planning based on cognitive maps. With cognitive maps, behaviors rely not only actions but also goals that will generate an activity propagated from action to action in order to trigger specific sequences. Adapting the behavior may not only be changing what action should be done but also changing what sequence is to be done. Determining which goal and thus which sequence is followed needs a particular processing due to the properties of the used cognitive maps. Potential goals are selected to estimate whether or not they are related to the current behavior of the robot. The selected goal is inhibited and then can be determined as the current goal if this inhibition eventually modifies the value of the propagated gradient that biases the activity of the selected action.

The architecture for planning with a cognitive map mainly relies on models of the Hippocampus, the Prefrontal Cortex and Parieto-temporal Cortices. The encoded low level actions are transitions corresponding to changes between two place-cells. The Hippocampus with the Entorhinal Cortex, known as a novelty detector can detect these changes of states. The cells of the Dendate Gyrus (DG) provide a time basis to the cells of the CA3 to predict the place-cell activities and thus events like changes of most recognized place-cell. These predictions are then used to predict transitions (in CA1) [12]. A competition between the possible transitions is performed at the level of the Basal Ganglia giving the action selection. It can be biased by the Prefrontal activities from the propagation in the cognitive maps.

This described system is not the only solution to explain how behavior can be adapted from negative feedback. It more likely corresponds to a last developmental stage of action planning. At first the brain can directly use simple sensorimotor actions valuated by a reinforcement learning process [7] occurring in the Basal Ganglia. In order to capture the correct properties of the task to be performed, this process must slowly adapt the encoding and thus the behavior. The Basal Ganglia can count on the frontal cortex to solve this problem. There exist several cortico-striatal loops involving the Basal Ganglia and the frontal cortex with different functional levels [13] [14]. The simple action planning directly based on reinforcement learning correspond to the motor loop represented in the frontal cortex by the Supplementary motor areas (SMA), the Premotor Cortex (PMC) and the Somatosensory area (SSA). When a negative signal is received, working memories [15] present in the frontal cortex could come to temporarily inhibit the incorrect actions. As a result, the behavior is adapted fast while the reinforcement learning process learn what to do at its own speed. A more cognitive loop (called spatial loop in [14]) includes the dorso-lateral cortex (DLC) and the posterior parietal cortex (PPC). This loop correspond to the cognitive map model. The goals would be in the dorso-lateral cortex whereas the cognitive map would be encoded in the recurrent connections of the posterior parietal cortex.

The nodes corresponding to motor actions in the motor loop find their homologous in the goals of the spatial loop. The difference is that these goals can propagate activities in networks (cognitive maps) thus encapsulating complete sequences of actions. Considering that the spatial loop are a development of the original motor loop, the same inhibition process can occur. Depending on the reception of a negative signal, the working memories can come to inhibit goals as well as actions providing the basis for enabling an agent to detect its own goal while planning with the cognitive map. In [15], the authors showed that cognitive tasks (Wisconsin test, Tower of London test) could be solved by neural network models of the prefrontal functions based on testing and selecting code-rule clusters (goals).

Implementing the motor and spatial loops in parallel can be used to get the best of the two strategies [16]. However their interactions may not be restricted to selecting which strategy is the best at a given moment. As the cognitive map can plan sequences of actions, such sequences could be reencoded as action primitives. The reinforcement learning process and the motor loop could directly process such primitives. As these primitives are new possible actions, they should be integrated in the cognitive representations of the possible actions i.e. in the cognitive maps. Then, the spatial loop could build sequences including the more complex action primitives. The development of more and more complex behaviors would not be performed by one superior structure but rather from the recurrent interactions between the quite simple motor loop and spatial loop. In order to handle these primitives and complex sequences, the nodes (representing goals or actions) should be reencoded as chunks merging adequately many different sensory signal [17]. An adequate extraction of the relevant features to be encoded is the challenge to be tackled [18].

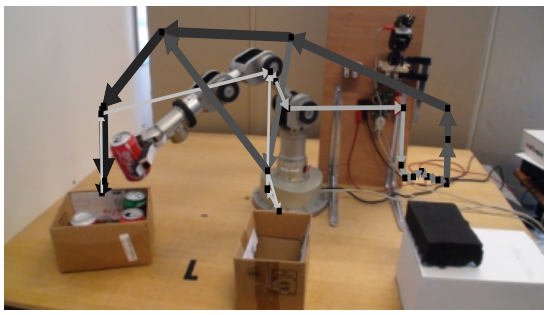


Fig. 8. Can sorting based on a cognitive map model: Pick and place experiment with a Katana (Neuronics AG) robotic arm.

Finally, current ongoing work also focuses on implementing and validating the own goal detection model on real robot (Fig. 8). Correctly taking negative feedback into account should improve how natural interactions can be, including in non-navigational tasks like arm control.

ACKNOWLEDGMENT

This work was supported by the INTERACT french project referenced ANR-09-CORD-014 and the NEUROBOT french ANR project.

REFERENCES

[1] S. Feinman, "Social referencing in infancy," *MerrillPalmer Quarterly*, vol. 28, no. 4, pp. 445–470, 1982.

[2] A. Thomaz, M. Berlin, and C. Breazeal, "An embodied computational model of social referencing," in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*. Nashville, TN, USA: IEEE, 2005, pp. 591–598.

[3] S. Boucenna, P. Gaussier, L. Hafemeister, and K. Bard, "Autonomous Development of Social Referencing Skills," in *From Animals to Animats 11*. Springer-Verlag Berlin, Heidelberg 2010, 2010, vol. 6226, pp. 628–638.

[4] C. Hasson, S. Boucenna, P. Gaussier, and L. Hafemeister, "Using Emotional Interactions for Visual Navigation Task Learning," in *Proceedings of the International Conference on Kansei Engineering and Emotion Research*, Paris France, 2010, pp. 1578–1587.

[5] C. Giovannangeli, P. Gaussier, and G. Désilles, "Robust Mapless Outdoor Vision-based Navigation," in *IEEE/RSJ International Conference on Intelligent Robots and systems*. Beijing, China: IEEE, 2006.

[6] C. Giovannangeli and P. Gaussier, "Interactive Teaching for Vision-Based Mobile Robots: A Sensory-Motor Approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 40, no. 1, pp. 13–28, 2010.

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 1998.

[8] N. Cuperlier, M. Quoy, C. Giovannangeli, P. Gaussier, and P. Laroque, "Transition Cells for Navigation and Planning in an Unknown Environment," in *From Animals to Animats 9*. Springer Berlin Heidelberg, 2006, vol. 4095, book part (with own title) 24, pp. 286–297.

[9] C. Giovannangeli and P. Gaussier, "Autonomous Vision-Based Navigation: Goal-Oriented Action Planning by Transient States Prediction, Cognitive Map Building, and Sensory-Motor Learning," in *IEEE/RSJ International Conference on Intelligent Robots and systems (IROS'08)*, 2008.

[10] P. Gaussier, A. Revel, J. P. Banquet, and V. Babeau, "From view cells and place cells to cognitive map learning: processing stages of the hippocampal system," *Biological Cybernetics*, vol. 86, no. 1, pp. 15–28, Jan. 2002.

[11] A. de Rengerve, J. Hirel, P. Andry, M. Quoy, and P. Gaussier, "On-line learning and planning in a pick-and-place task demonstrated through body manipulation," in *IEEE International Conference on Development and Learning (ICDL) and on Epigenetic Robotics (Epirob), 2011*, vol. 2, Aug. 2011, Journal article, pp. 1–6.

[12] J. Hirel, P. Gaussier, and M. Quoy, "Biologically inspired neural networks for spatio-temporal planning in robotic navigation tasks," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, dec. 2011, pp. 1627–1632.

[13] G. E. Alexander, M. R. DeLong, and P. L. Strick, "Parallel organization of functionally segregated circuits linking basal ganglia and cortex." *Annual Review of Neuroscience*, vol. 9, no. 1, pp. 357–381, 1986.

[14] A. D. Lawrence, B. J. Sahakian, and T. W. Robbins, "Cognitive functions and corticostriatal circuits: insights from Huntington's disease," *Trends in Cognitive Sciences*, vol. 2, no. 10, pp. 379–388, Oct. 1998. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1364661398012315

[15] S. Dehaene and J. P. Changeux, "Neuronal models of prefrontal cortical functions." *Annals Of The New York Academy Of Sciences*, vol. 769, no. 1 Structure and Functions of the Human Prefrontal Cortex, pp. 305–319, 1995.

[16] L. Dollé, D. Sheynikhovich, B. Girard, R. Chavarriaga, and A. Guillot, "Path planning versus cue responding: a bio-inspired model of switching between navigation strategies," *Biological Cybernetics*, vol. 103, no. 4, pp. 299–317, Oct. 2010.

[17] G. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956.

[18] S. Hanoune, M. Quoy, and P. Gaussier, "An architecture for online chunk learning and planning in complex navigation and manipulation tasks," in *IEEE International Conference on Development and Learning (ICDL) and on Epigenetic Robotics (Epirob), 2012*, 2012, Journal article, p. Submitted.

3.2.2 Exploitation des signaux de renforcement négatifs pour modifier le comportement du robot dans l'expérience de tri de canettes

Le modèle de carte cognitive avec gestion des buts a ensuite été implémenté pour utiliser les retours négatifs dans l'expérience de tri des canettes sur un robot réel (Fig. 3.2). L'expérience consiste toujours à trier deux types de canettes (verte ou rouge) et à les placer dans deux boites positionnées à deux endroits différents dans l'environnement. Les séquences de mouvements sont apprises à la volée. Il y a essentiellement un point de bifurcation des trajectoires lorsque le robot doit déterminer dans quelle boite il va déposer la canette. Le robot suit le même protocole d'apprentissage du tri que dans la section 3.1 (démonstration passive des différentes séquences de gestes et renforcement positif lorsque la canette est lâchée dans la bonne boite). Puis, nous montrons au robot qu'il doit déposer les canettes rouges à un autre endroit en manipulant physiquement le bras de robot et en lui signalant le succès de la démonstration par un retour positif (récompense). Comme la précédente association avait été renforcée plusieurs fois, le robot maintient son comportement erroné. Nous pouvons alors utiliser un signal négatif pour permettre le désapprentissage. L'association contexte-but erronée est diminuée, permettant à l'autre but de se propager jusqu'au niveau de la bifurcation. Cependant, le robot n'avait pas appris de chemin lui permettant d'interrompre son geste après le point de bifurcation. Il doit donc d'abord terminer la séquence en cours. La fois suivante, le robot réalise la bonne séquence de gestes. En utilisant des signaux sociaux valués (positifs ou négatifs), l'humain a donc pu modifier le comportement du robot sans entrer dans une interaction physique fastidieuse avec celui-ci.

3.3 Conclusion - discussion

Dans ce chapitre, nous avons étendu le modèle DM-PerAc (chapitre2) en incluant des mécanismes d'apprentissage de séquences et de buts issus de modèles liés à la navigation afin d'obtenir l'apprentissage de tâches impliquant des objets ou des interactions sociales. Le modèle complet s'appuie sur les interactions entre différentes structures cérébrales modélisées (cortex moteur et préfrontal, hippocampe, cortex préfrontal, ganglions de la base et cervelet). Les associations visuomotrices au niveau du cortex prémoteur ont permis d'apprendre un contrôleur définissant un homéostat visuomoteur. Combiné à l'ambiguïté de la perception, ce contrôleur permet à un robot d'imiter en simultané des gestes observés. Le modèle hippocampique permet d'apprendre des transitions entre deux états catégorisés, par exemple visuels. Nous avons supposé que le robot est aussi capable d'inhiber ses actions motrices lors d'une démonstration par le professeur. Ainsi, grâce au contrôleur visuomoteur appris, le robot peut reproduire une séquence observée en différée [1]. Le robot a appris par observation. Pour des tâches plus complexes, les transitions apprises sont utilisées pour apprendre, au niveau du cortex pariétal et préfrontal, un graphe des séquences possibles suivant le principe des cartes cognitives utilisées en navigation. Cet apprentissage permet au robot d'associer des contextes motivationnels et des buts avec des séquences de gestes. Une retour social non-verbal valué, tel qu'un sourire (positif) ou un froncement de sourcils (négatif), suffit alors pour modifier le comportement du robot et rendre l'interaction plus aisée pour l'humain.

1. Dans le cas de routines sensorimotrices, l'apprentissage du timing devrait s'effectuer au niveau du cervelet selon un mode très similaire à celui que nous avons simulé au niveau de l'hippocampe.

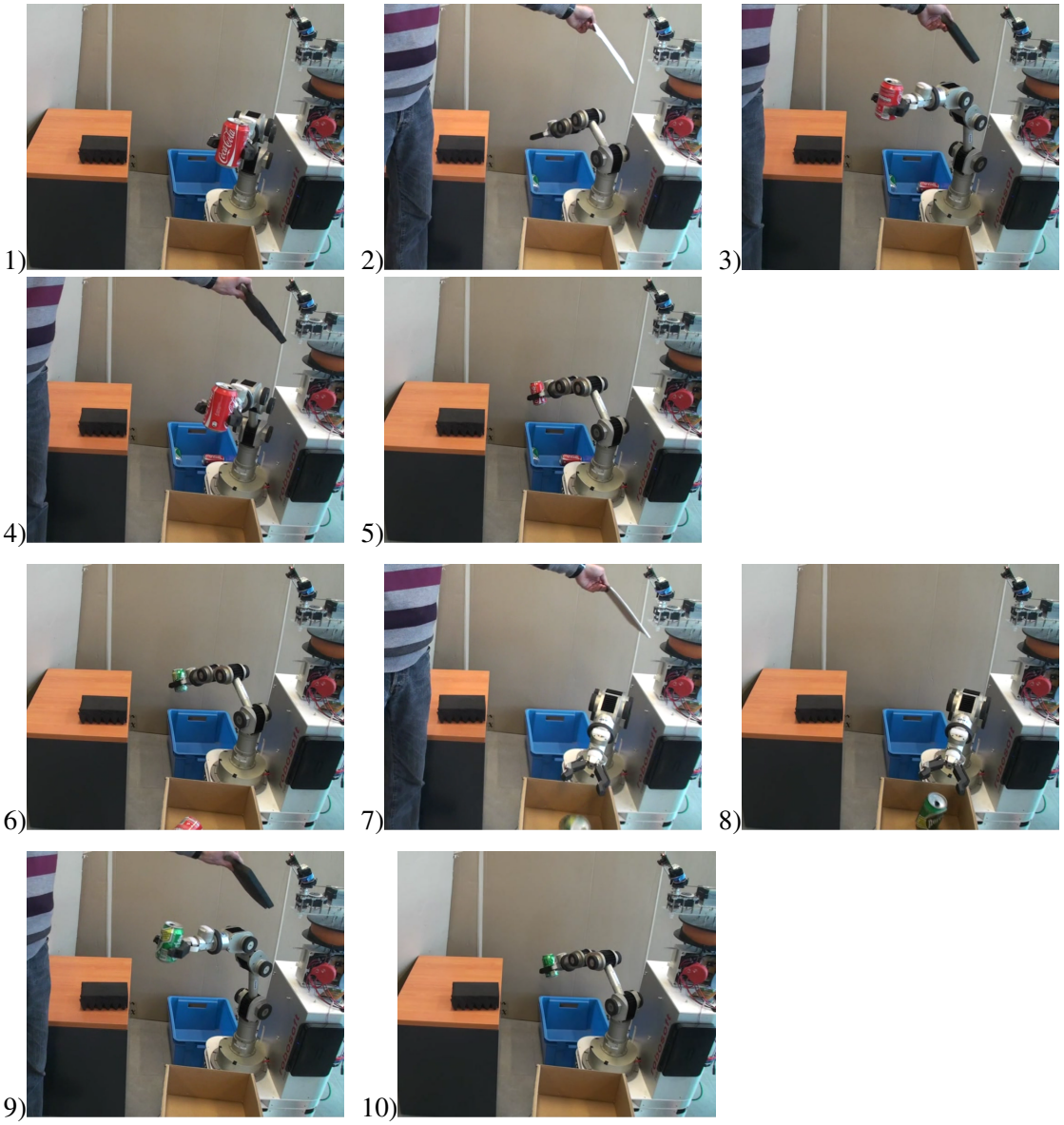


FIGURE 3.2 – Expérience de tri des canettes en incluant les retours négatifs de l'humain dans l'adaptation du comportement du robot. Les images sont numérotées de 1 à 10 dans l'ordre suivant : de gauche à droite puis de haut en bas. 1-5 : Adaptations concernant les canettes rouges. 1)Tri initial incorrect. 2) Démonstration du tri correct et retour positif. 3) Erreur du robot. Retour négatif. 4) Nouvelle erreur du robot. Retour négatif. 5) Le comportement a été adapté. 6-10 : Adaptation du comportement lié aux canettes vertes. 6) Tri initial incorrect. 7) Démonstration du tri correct et retour positif. 8) Comportement correct. 9) Retour au comportement initial, le nouveau comportement ne doit plus être reproduit. Retour négatif donné au robot. 10) Le comportement final est bien le comportement initial.

Le modèle proposé est basé sur des modèles développés en navigation sans lien avec l'aspect social. Nous avons réalisé une transposition à un cadre social qui peut devenir une base pour progresser vers des comportements plus collaboratifs. Les capacités d'imitation différée donnent gratuitement une forme simple de reconnaissance des intentions. Dans l'article 2, figure 8, la séquence de mouvements du robot est déclenchée par la position de la main de l'humain qui est la position initiale de la séquence précédemment montrée. Le robot n'est pas capable d'atteindre cette position car elle est hors de son espace d'action. Le robot ne réalise que la partie de la séquence qui correspond à ses capacités motrices. Pour un observateur, l'humain esquisse le geste vers l'objet et le robot, comme s'il avait détecté l'intention de l'humain, vient compléter la séquence démarrée par l'humain. Suivant le modèle décrit à la section 3.2, les renforcements positifs ou négatifs peuvent modifier le comportement du robot en venant créer, renforcer ou inhiber des buts. Ces buts peuvent être considérés comme l'intention motrice du robot. L'avantage de ce système est donc de permettre de modifier le comportement du robot à travers son intention motrice avec un minimum d'interaction. Ainsi, l'humain n'est pas obligé d'interrompre son travail pour remontrer la tâche que le robot doit réaliser. [Tomasello et al., 2005] identifie trois caractéristiques-clés pour la collaboration : (1) les agents se répondent mutuellement et sont coordonnés dans leur actions, (2) ils ont un plan d'action en commun pour leur entreprise conjointe (ce plan conjoint se confirme par la capacité des agents à inverser les rôles) et (3) ils font preuve d'un engagement mutuel pour réaliser les actions participant au but commun. Si l'imitation immédiate permet une réponse mutuelle des agents, celle-ci est absente dans le cas de la tâche de tri des canettes. Il ne s'agit en fait pas d'une tâche impliquant un partage des actions suivant un plan conjoint et il n'y a donc pas de rôle à inverser. En pratique, le robot apprend et réalise le rôle (la séquence) que l'on veut bien lui montrer. Cependant, la possibilité de construire et d'adapter les buts poursuivis suivant le retour social permet de guider le robot pour qu'il s'engage sur l'objectif désiré qui peut alors devenir commun.

Dans la tâche de tri des canettes, le robot construit une carte cognitive incomplète. En pratique, elle contient les différentes séquences qui seront utiles pour la réalisation de la tâche. Une carte cognitive complète n'est pas nécessaire. De plus, elle impliquerait l'existence de nombreux raccourcis par rapport aux séquences montrées. Ces raccourcis peuvent ou non être pertinents. La carte cognitive permet de suivre le plus court chemin jusqu'au but le plus proche et le plus actif. Pour obtenir des séquences particulières avec une carte cognitive complète, nous envisageons deux solutions possibles : apprendre des sous-buts qui justifient le détour ou inhiber les transitions incorrectes. En forçant le potentiel des transitions incorrectes à zéro, le gradient n'est plus diffusé par ces transitions. Ainsi, le gradient suivi correspondra à un séquence évitant les transitions inhibées. [Hirel, 2011] propose un modèle permettant d'inhiber les transitions qui échouent i.e. qui malgré leur sélection ne sont pas réalisées d'un point de vue perceptif (changement d'état). Ce mécanisme peut être conservé tel quel dans le cadre d'un apprentissage par démonstration. En effet, durant la démonstration, l'humain peut contraindre le robot à réaliser certaines transitions particulières plutôt que celles prédites par la carte cognitives : les transitions qui échouent sont celles que le système ne doit pas réaliser. Dans ses expériences, [Hirel, 2011] utilise une mémoire des transitions inhibées qui est limitée dans le temps. Ces inhibitions pourraient être apprises à plus long terme et associées avec un contexte, a priori le contexte motivationnel actif.

Dans la tâche de tri de canettes, le robot ne tient pas compte de la position des boites pour réaliser le tri. L'objet est relâché à la position apprise indépendamment de la configuration de

l'environnement. Le modèle ne gère pas actuellement le cas d'un tri où l'on associe les canettes à des boites particulières plutôt qu'à des positions dans l'espace. Dans cet exemple, le robot devrait adapter ses mouvements non seulement en fonction de la couleur de la canette à trier mais aussi en fonction de la position de la boite dans laquelle le robot doit déposer la canette. A partir du modèle étudié dans ce chapitre, nous voyons deux approches pour résoudre cette tâche. La première approche consiste à adapter la stratégie de contrôle en fonction de la tâche à réaliser. En effet, si le robot doit déposer la canette au dessus d'une boite particulière, il serait préférable d'avoir une stratégie qui s'appuierait sur l'information visuelle plutôt que l'information proprioceptive. Il ne serait alors plus nécessaire d'apprendre une séquence. Pour cela, il suffirait que l'attention soit dirigée vers la position visuelle souhaitée et que le contrôleur visuomoteur utilise cette information visuelle pour guider les mouvements vers la configuration motrice pour déposer l'objet au bon endroit. Une difficulté majeure est alors la sélection de la bonne stratégie. Nous avons vu différents types de comportements (imitation versus planification des gestes, séquences visuelles versus séquences proprioceptives) qui peuvent tous se révéler pertinents suivant les situations rencontrées. Dans mes travaux de thèse, le choix de la stratégie a été fait manuellement. Des travaux comme [Dollé et al., 2010; Nguyen and Oudeyer, 2012] et [Jauffret et al., 2013] apportent des éléments de solutions pour la sélection de stratégies. Dans [Dollé et al., 2010], le modèle apprend à sélectionner entre deux stratégies de navigation (basée sur des repères visuels (stimulus-réponse) versus basée sur des cellules de lieu (carte cognitive)). Un apprentissage par renforcement [Sutton and Barto, 1998] permet de choisir la stratégie dont la direction prédite correspond le mieux à la direction du but à rejoindre. [Jauffret et al., 2013] proposent que chaque stratégie s'auto-évalue suivant le comportement "normal" qu'elle s'attend à générer. Les prédictions sensorielles associées à une stratégie sont comparées à différents ordres avec les sensations actuelles. La détection d'une irrégularité par rapport à la prédiction conduit à l'inhibition de la stratégie correspondante, permettant que seules les stratégies dont les prédictions sont fiables puissent d'influencer le comportement. Dans [Nguyen and Oudeyer, 2012], les auteurs proposent une algorithme permettant à un robot d'apprendre quelle stratégie d'apprentissage utiliser en fonction de l'état du système. Les stratégies d'apprentissage considérées sont une exploration autonome active[Baranes and Oudeyer, 2010], une émulation d'un professeur sélectionné activement parmi les professeurs disponibles et une reproduction mimétique des gestes d'un professeur sélectionné. La capacité à sélectionner de manière autonome la stratégie la plus appropriée permet de réduire le temps d'apprentissage et offre une solution complémentaire à notre approche.

La seconde approche pour obtenir un tri dépendant de la position de la boite est de catégoriser plus précisément la situation et l'état du robot. En effet, si le comportement à réaliser doit dépendre de la couleur de la canette et de la position des boites alors le contexte motivationnel pourraient encoder ces deux informations en même temps. La difficulté est que toute modalité peut potentiellement être pertinente. Plus il y a de modalités prises en compte plus la reconnaissance d'une situation et sa généralisation deviennent difficiles à maîtriser et à apprendre (malédiction de la dimension, "curse of dimensionality" [Bellman, 1957]). Chaque modalité peut avoir une influence différente sur la reconnaissance d'une situation et cette influence devrait être apprise. Actuellement, notre modèle ne gère pas ce type d'apprentissage. Le choix et la pondération, a priori, des modalités utilisées dans ces différentes catégories a simplifié fortement l'apprentissage. Par exemple, le fait que l'information de la couleur de la canette est utilisée dans les contextes motivationnels et pas pour créer les transitions permet de simplifier l'apprentissage

de la tâche de tri des canettes. La question des pondérations des modalités apparaît dans l'apprentissage des catégories sensorielles (visuelles, proprioceptives, ou visuomotrices) qui peuvent être des contextes motivationnels ou des états sur lesquels sont construits les transitions. Les algorithmes LWPR et GMM/GMR (Sec. 1.1.3 et 1.1.3) peuvent apprendre l'influence des différentes modalités via l'adaptation des coefficients de covariance qui déterminent la sélectivité des noyaux gaussiens utilisés. De plus, l'algorithme LWPR peut apprendre à projeter les données pour ignorer certaines dimensions non pertinentes. Cependant, ces apprentissages s'appuient sur de multiples exemples de trajectoires pour extraire l'importance de chaque modalité. Quelles modalités ou informations doivent être utilisées pour quelles catégories sensorielles et surtout comment cela peut-il être appris en ligne ? Cette problématique sera essentielle pour réaliser l'apprentissage en ligne d'une tâche mêlant des comportements différents comme naviguer et contrôler une bras robotique. En effet, la sélection des comportements devra dépendre de multiples modalités pouvant être liées à l'un ou à l'autre des comportements considérés. Dans le chapitre suivant, nous nous intéresserons à l'intégration dans une même tâche de comportements de navigation et de contrôle d'un bras. Le robot devra se déplacer entre différents lieux pour prendre ou déposer des objets. De par le nombre de modalités mises en jeu et les différentes décisions (déplacements, manipulation) impliquées, cette tâche sera très intéressante pour étudier l'apprentissage de la sélection d'action sur la base d'une catégorisation multimodale des situations rencontrées. Nous aborderons la création et l'utilisation des catégories multimodales pour la sélection d'action lors d'un apprentissage en ligne de tâches avec un professeur humain.

Publications personnelles

de Rengervé, A., Boucenna, S., Andry, P., and Gaussier, P. (2010a). Emergent Imitative Behavior on a Robotic Arm Based on Visuo-Motor Associative Memories. In *IEEE/RSJ International Conference on Intelligent Robots and systems (IROS'10)*, pages 1754–1759, Taipei, Taiwan

de Rengerve, A., Hirel, J., Andry, P., Quoy, M., and Gaussier, P. (2011). On-line learning and planning in a pick-and-place task demonstrated through body manipulation. In *IEEE International Conference on Development and Learning (ICDL) and on Epigenetic Robotics (Epirob), 2011*, volume 2, pages 1–6, Frankfurt am Main, Germany. IEEE

Dans toute action, dans tout choix, le bien c'est la fin, car c'est en vue de cette fin qu'on accomplit toujours le reste.

– Aristote

CHAPITRE 4

Utilisation de contextes multimodaux pour la sélection de comportements dans une tâche mêlant navigation et contrôle d'un bras

Nous avons vu au chapitre 1 que des apprentissages sensorimoteurs simples (approche PerAc, sec. 1.4) permettait de construire des attracteurs comportementaux (homing, suivi de route...). Cette approche a été étendue au contrôle visuomoteur d'un bras robotique (chapitre 2) qui nous a ensuite servi pour mettre en place des comportements d'imitation et d'apprentissage de tâches de plus en plus complexes (chapitre 3). Notre robot peut maintenant exhiber différents comportements. Dans ce chapitre, nous allons aborder le problème de la sélection de comportements dans le cadre de tâches mêlant navigation et contrôle d'un bras (exemple en figure 4.1). Ces tâches

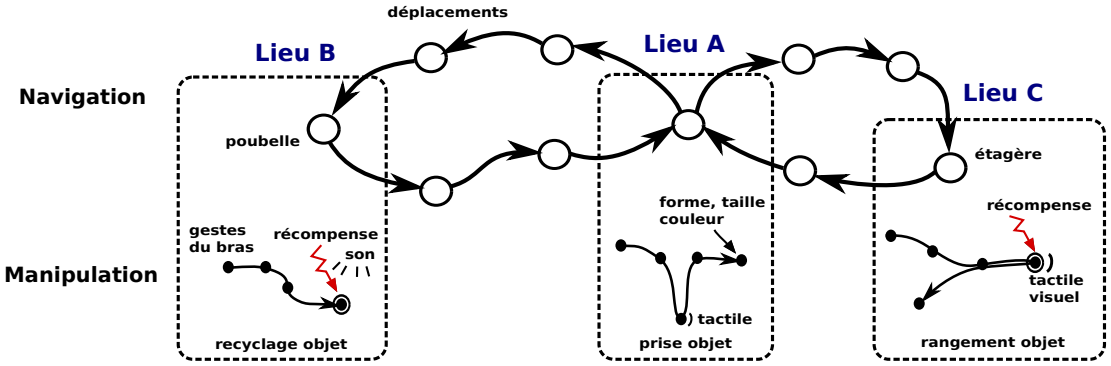


FIGURE 4.1 – Exemple de tâche mêlant navigation et manipulation d'objets. Les déplacements ou les gestes du bras sont sélectionnés en fonction d'informations multimodales (localisation dans l'espace, son, tactile, posture du bras, détection d'un objet à ranger, etc.). Par exemple, le robot doit s'appuyer sur les informations liées à l'objet (taille, forme,...) pour déterminer dans quel lieu se rendre ensuite.

nous intéressent car elles présentent une grande richesse en terme de modalités impliquées (localisation spatiale, proprioception du bras, retour de capteurs ultrason détectant des obstacles, retour tactiles, etc.) et de comportements pouvant être réalisés (naviguer vers différents lieux, attraper des objets, les manipuler, les déposer...). Ces comportements correspondent à des actions ou des séquences d'actions plus ou moins complexes qui devront être sélectionnées et réalisées en fonction des situations décrites par les différentes modalités. Typiquement, une tâche peut être

de trier et ranger des objets dans différents endroits en fonction de leur catégorie (figure 4.1). Dans notre cas, le robot devra aller récupérer des objets à un endroit A pour les déposer en B ou les jeter en C en fonction de la reconnaissance de leur catégorie (ici la taille de l'objet pour simplifier). Une fois l'objet tenu déposé, le robot reprend sa ronde pour récupérer d'autres objets. Dans cette tâche, les différents comportements doivent être réalisés dans un ordre correct, par exemple, notre robot doit évidemment tenir un objet avant de pouvoir l'emporter avec lui. La difficulté sera de faire apprendre au robot la sélection de la bonne tâche (ou dit plus précisément la sélection du bon attracteur) en fonction des directives fournies par son professeur durant des interactions non-verbales.

A partir du milieu des années 80, les solutions au problème de sélection d'actions basées sur l'exécution des étapes d'un plan fixé a priori ont été progressivement abandonnées pour l'utilisation de comportements réactifs [Brooks, 1986; Arkin, 1998]. Ces modules peuvent surveiller en continu l'environnement permettant au système de réagir plus rapidement aux changements de l'environnement. Les modules comportementaux peuvent représenter des actions de plus ou moins haut niveau (e.g. mouvement d'une patte [Maes and Brooks, 1990] ou comportement de reproduction [Tyrrell, 1993; Bryson, 2000] pour des agents artificiels imitant les comportements d'animaux). Les modules de comportements peuvent être organisés en parallèle [Brooks, 1986; Maes and Brooks, 1990] ou suivant une hiérarchie [Tyrrell, 1993]. [Bryson, 2012] décrit l'approche intermédiaire. Bryson défend l'idée qu'une activation partielle de modules réactifs suivant une hiérarchie soigneusement définie, avec la capacité de basculer l'attention sur d'autres branches de la hiérarchie, constitue une solution robuste qui facilite la coordination des comportements. Dans [Brooks, 1986], la coordination est basée sur l'architecture dite de *subsumption* qui définit, a priori, la priorité pour chaque comportement. Maes et Brooks [Maes and Brooks, 1990] ont réalisé l'apprentissage de la coordination des comportements sur un robot à 6 pattes qui apprend à marcher. Chaque comportement (mouvement d'une patte) est déclenché par sa conjonction de conditions (estimateurs binaires sur le monde, e.g. teste si une patte donnée est en l'air). Cette conjonction est mise à jour au cours de l'apprentissage pour favoriser la sélection des comportements fiables et pertinents. Initialement, la conjonction de conditions est (presque) vide, le comportement est souvent activé. Des variables statistiques mesurant l'obtention d'un renforcement (positif ou négatif) avec ce comportement, en fonction des conditions (vraies ou fausses) permettent de déterminer quelles conditions inclure dans la conjonction. L'activation d'un comportement devient ainsi de plus en plus sélective afin d'obtenir la bonne coordination des comportements. Les contextes appris (conjonction de conditions) déterminent quels comportements sélectionner. Dans [Thrun and Mitchell, 1995], Thrun et Mitchell insistent sur l'importance de réutiliser les comportements déjà existants pour transférer le savoir sur les tâches plus complexes. Dans un premier temps, le système apprend des modèles prédisant les conséquences des comportements. Ensuite, lors de l'apprentissage d'une tâche complexe par Q-learning [Watkins and Dayan, 1992], les contrôleurs déjà appris peuvent être utilisés pour accélérer l'apprentissage de la fonction de valeur.

Pour commencer, dans la section 4.1, nous montrerons l'avantage d'avoir des comportements qui s'expriment en même temps pour obtenir de manière émergente des comportements plus sophistiqués. Par exemple, si deux boucles de contrôle simples sont exécutées en parallèle, l'une pour le bras robotique et l'autre pour l'orientation du robot, sans synchronisation ni planification particulière, le robot se déplacera en tendant son bras en direction de son but. Lorsque le but est d'attraper une balle que se lancent deux humains, on constate que le robot se comporte

à la manière d'un enfant avec une dynamique de comportement unique et très lisible comme si un seul comportement global avait été programmé.

Malheureusement cette approche ne peut pas être généralisée à un ensemble arbitraire de comportements. Il est nécessaire que les comportements puissent être réalisés indépendamment les un des autres. Par exemple, le robot ne doit pas changer de direction pour éviter un obstacle alors que l'on cherche à attraper un objet posé sur cet obstacle (une boite posée sur une table par exemple). Notre robot devra donc apprendre quels comportements peuvent être réalisés ou quels sont ceux qui doivent être inhibés. Une solution classique pour la sélection de comportements est un apprentissage par renforcement comme le Q-Learning [Watkins and Dayan, 1992] (Figure 4.2a). Cependant, un tel apprentissage est relativement lent et ne correspond donc pas à notre objectif d'un apprentissage réactif en ligne. Nous nous intéresserons à un usage de contextes recrutés permettant d'apprendre rapidement quels comportements sélectionner.

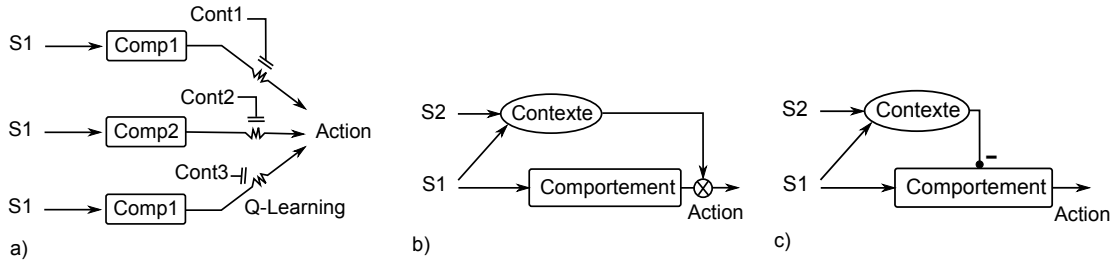


FIGURE 4.2 – Modèles d'apprentissage de sélection de comportements. (a) Les actions prédites par chaque comportement sont modulées par un contexte avant d'être fusionnées. Les pondérations des différents comportements peuvent être apprises par renforcement (Q-learning). Cet apprentissage est relativement lent. (b) Le contexte recruté vient simplement inhiber de manière momentanée un comportement permettant aux autres comportements de s'exprimer. Il y a un nombre potentiellement infini de conditions dans lesquels un comportement peut être actif. (c) Un contexte recruté vient sélectionner en positif le comportement à réaliser, suivant un effet multiplicateur. Le contexte décrit donc une condition de fonctionnement du comportement.

Nous nous intéresserons d'abord à l'apprentissage de contextes venant inhiber de manière ponctuelle certains comportements (Figure 4.2b). Dans ce modèle, le comportement est actif tant qu'aucun contexte l'inhibant n'est activé. Il y a alors potentiellement un nombre infini de conditions dans lesquelles le comportement sera réalisé, donnant plus de liberté pour la sélection d'action. L'évaluation et la détection des comportements à inhiber est difficile à réaliser directement. Avec l'approche PerAc (Section 1.4), nous avons vu que les comportements émergeant des bassins d'attraction générés par les couples sensorimoteurs appris. Nous proposons donc que l'évaluation et la sélection des comportements soit faite à partir de l'évaluation et de la sélection des couples sensorimoteurs durant l'interaction avec le professeur humain. Cette interaction guidant le comportement du robot peut être alors utilisée pour apprendre des contextes venant inhiber de manière ponctuelle certains couples sensorimoteurs. Dans la section 4.2, nous appliquerons ce principe à l'apprentissage interactif de trajectoires dépendant d'un contexte (e.g. objet tenu). Les couples sensorimoteurs devant être inhibés seront détectés en comparant les attentes liées au modèle avec l'action du professeur humain lorsqu'il corrige le comportement du robot. Notre modèle mémorisera alors ces couples sensorimoteurs en les associant avec des contextes multimodaux. Nous montrerons en simulation que le bassin d'attraction généré est bien modulé par les contextes appris : le robot suit des trajectoires différentes en fonction de l'objet

tenu.

Nous étudierons ensuite le cas où le contexte décrit directement la condition de fonctionnement du comportement (Figure 4.2c). Les contextes, recrutés quand le robot fait une erreur, pondèrent les comportements avec un effet multiplicateur : un comportement ne sera réalisé que si un contexte qui l'active est reconnu. Dans la section 4.3, nous présenterons une expérience dans laquelle le robot devra réaliser la tâche de tri avec déplacement décrite (figure 4.1). Les différents comportements (navigation, contrôle du bras, contrôle de la pince pour prendre ou relâcher des objets) seront associés avec des contextes multimodaux venant les activer. Une compétition sur les comportements déterminera celui qui est réalisé. Par ailleurs, l'apprentissage, en ligne, utilisera les démonstrations d'un professeur humain guidant le robot. Notre modèle permettra ainsi au robot de reproduire la tâche de tri avec déplacement.

Enfin, dans la section 4.4, nous discuterons les perspectives en terme d'apprentissage développemental de la sélection de comportements. En particulier nous reviendrons sur l'importance d'apprendre des contextes multimodaux qui fusionnent les informations essentielles afin d'améliorer les capacités de généralisation.

4.1 La coopération de comportements simples permet d'obtenir des comportements complexes

Nous allons montrer que des comportements qui s'exécutent en parallèle peuvent résoudre des tâches complexes, difficiles à planifier autrement. Nous avons implémenté sur un robot mobile équipé d'un bras robotique et d'une caméra mobile une architecture qui exécute en parallèle le contrôle du bras robotique et le contrôle de l'orientation (`http://perso-etis.ensea.fr/~neurocyber/Videos/mobileManipulation/reactivController.mpg`). L'attention visuelle du robot est focalisée sur un objet saillant (une balle), détectée par exemple grâce à sa couleur. La position de la balle dans le champ visuel est projetée dans un espace angulaire invariant par rapport aux mouvements de la caméra (référentiel angulaire corps-centré). La comparaison entre l'orientation actuelle de la caméra et l'orientation de la balle dans cet espace donne la correction à appliquer à l'orientation de la caméra pour que celle-ci suive la balle.

Le contrôle du bras robotique est basé sur le contrôleur visuomoteur mis en place au chapitre 2. Dans un premier temps, le robot apprend la coordination visuomotrice entre la position visuelle de sa pince et la posture de son bras. Durant l'expérience, l'information visuelle en entrée du contrôleur provient de la position visuelle de la balle. Le bras robotique est donc déplacé dans une configuration motrice qui permet d'aligner visuellement la pince et la balle.

Le contrôle de l'orientation du robot est décrit par la figure 4.3. Un champ de neurones dynamiques est utilisé pour fusionner les orientations proposées par trois stratégies différentes : une stratégie d'exploration aléatoire, la réorientation du robot vers la balle suivant sa position dans l'espace corps-centré, et la réorientation du robot sur la direction donnée par un joystick simulant la possibilité pour un humain de pousser le robot dans une direction particulière. Ces stratégies ont été énoncées suivant leur ordre de priorité, de la plus basse à la plus haute. Le robot peut aussi explorer aléatoirement un environnement ou être guidé par un humain. Nous nous intéresserons ici plus particulièrement au cas où l'orientation du robot est déterminée par la position de la balle. A chaque instant, un mécanisme d'évitement d'obstacles utilisant les capteurs à ultrason disposés sur la base mobile permet de modifier l'orientation du robot afin qu'il évite les obstacles. La vitesse linéaire du robot dépend de la proximité des obstacles mais

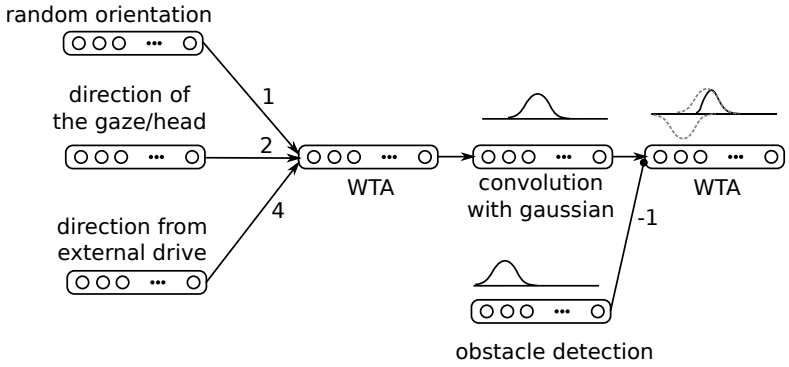


FIGURE 4.3 – Architecture déterminant l'orientation du robot.

aussi de la proximité de la balle, estimée par sa taille perçue (le nombre de pixels dans l'image associés à la balle).

L'exécution de ces comportements, sans synchronisation ni planification particulière, suffit pour que notre robot réalise le comportement d'attraper une balle (Figure 4.4). Lorsque le robot avance vers la balle, il abaisse progressivement son bras pour finir par toucher la balle lorsqu'il est suffisamment près. Au fur et à mesure que la balle se rapproche, la hauteur à laquelle elle est perçue diminue, impliquant le mouvement du bras. Dans une deuxième expérience (même figure 4.4), la tâche du robot est d'attraper une balle que se lancent deux humains. On constate que le robot se comporte d'une manière très lisible (à la manière d'un enfant) comme si un seul comportement global avait été programmé. La possibilité de réaliser différents comportements en parallèle apparaît comme une propriété importante pour permettre la réalisation de comportements sophistiqués à moindre coût. De plus, cela permet d'avoir une adaptation très rapide aux variations de l'environnement, des comportements plus opportunistes, et une durée des calculs constante, simplifiant le problème du contrôle temps réel. Néanmoins, tous les comportements ne peuvent pas toujours s'exécuter en parallèle. Par exemple, une fois que le robot a attrapé l'objet, il doit cesser de s'orienter vers celui-ci pour aller vers l'endroit où le déposer. La question qui se pose alors est d'arriver à détecter les comportements inappropriés afin d'empêcher leur exécution.

4.2 Vers l'inhibition de comportements : détection, mémorisation et inhibition des couples sensorimoteurs indésirables

Il est difficile d'évaluer directement si un comportement est approprié ou non par rapport à la tâche considérée. Suivant l'approche PerAc (Sec. 1.4), un comportement est défini par le bassin d'attraction généré par les couples sensorimoteurs appris. L'évaluation d'un comportement peut donc résulter de l'évaluation des couples sensorimoteurs correspondant. Nous présentons dans cette section comment cette évaluation peut s'appuyer sur l'action d'un professeur humain corrigeant le comportement de notre robot, ainsi que l'utilisation de cette évaluation dans un apprentissage en ligne et interactif pour mémoriser les couples sensorimoteurs à inhiber. Grâce à cet apprentissage rapide et à cette inhibition, notre robot modifiera immédiatement son comportement, permettant au professeur de continuer à corriger les actions du robot si nécessaire.

Dans les travaux réalisés en navigation par [Giovannangeli and Gaussier, 2010] (cf. 1.4),

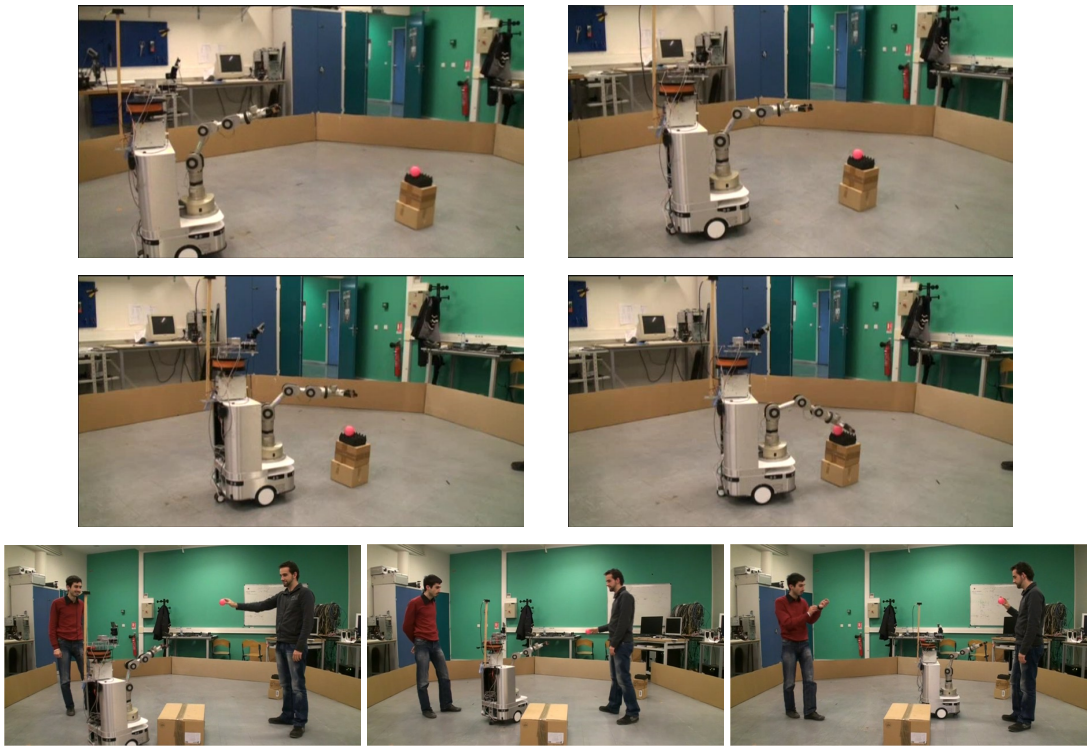


FIGURE 4.4 – Comportement d'attraper un balle initialement hors de portée du robot. (haut et centre) Le robot se déplace vers la balle et positionne son bras pour toucher la balle quand il est suffisamment près. La pince est maintenue dans l'alignement visuel de la balle. Le bras descend au fur et à mesure que le robot se rapproche, du fait de la perspective. (bas) Avec les mêmes comportements en parallèle, le robot peut aussi éviter les obstacles tout en essayant d'atteindre une balle tenue par un partenaire d'interaction humain. http://perso-etis.ensea.fr/~neurocyber/Videos/mobileManipulation/reactivController.mpg

l'apprentissage des cellules de lieu s'appuie sur la détection de l'action de l'humain qui modifie la direction prise par le robot (correction du comportement). Le robot compare la direction courante avec la direction prédite par son modèle interne basé sur des associations lieu-action. Si la direction prédite par son modèle est trop différente de la direction après correction, le robot déclenche alors l'apprentissage d'une nouvelle cellule de lieu permettant de compléter le modèle en incluant le comportement corrigé. Nous proposons d'utiliser ce type d'information (détection de l'action de l'humain par la comparaison de la situation avec la prédiction) pour apprendre les contextes multimodaux venant inhiber les comportements ne devant pas être réalisés. Plus précisément, c'est la dynamique de la perception qui doit permettre d'évaluer les orientations incorrectes. Initialement, les couples lieu/action appris déterminent la dynamique du mouvement pour réorienter le robot dans la direction prédite. Cependant l'action correctrice du professeur peut forcer un changement d'orientation qui s'oppose à cette dynamique i.e. l'orientation courante s'éloigne de l'orientation désirée (Fig. 4.5). Détecter et encoder ces situations permet au système de connaître les couples lieu/action à inhiber et d'apprendre les contextes multimodaux mémorisant cette inhibition. Le système s'appuie sur une base classique pour la navigation (i.e. avec des cellules de lieu issues de la combinaison d'amers-azimuts, Sec. 1.4).

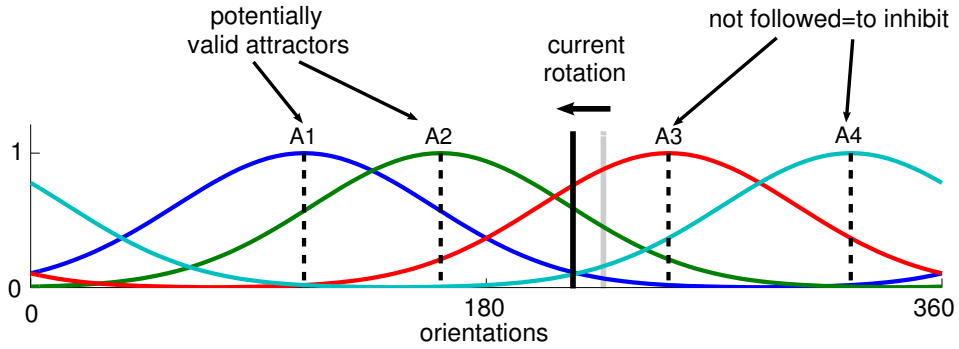


FIGURE 4.5 – Principe pour déterminer les attracteurs en orientation qui ne doivent pas être considérés. Les barres verticales pleines indiquent la position précédente (gris) et la position actuelle (noir). Suivant la rotation courante, le niveau dans la bulle d'activité de l'attracteur A3, par exemple, a diminué. Le mouvement s'oppose donc à la dynamique proposée par le champ d'attraction de A3. Traduit en terme de retour fourni par l'humain, cela signifie que l'attracteur A3 doit être inhibé.

Lorsque le comportement du système est corrigé, des contextes multimodaux peuvent être recrutés pour affiner les couples lieu/action activés. L'avantage d'utiliser les changements d'orientation pour évaluer les couples sensorimoteurs lieu/action est que l'apprentissage des contextes et des couples inhibés ne nécessite pas que la correction soit terminée et le robot correctement orienté. Les couples sensorimoteurs qui se révèlent inappropriés sont inhibés.

De nouveaux couples sensorimoteurs peuvent aussi être recrutés quand c'est nécessaire. Ce recrutement permet d'éviter une sur-généralisation des couples sensorimoteurs, et de résoudre les situations d'échec lorsque le même contexte ne suffit pas à apprendre correctement la sélection. Le mécanisme pour éviter la sur-généralisation est simple : si, lors de la correction, l'inhibition des couples sensorimoteurs aboutit à sélectionner des couples qui sont trop mal reconnus (i.e. reconnaissance du lieu en dessous du niveau de vigilance) alors un nouveau couple sensorimoteur est recruté. Dans la situation d'échec de l'apprentissage de la sélection à cause du contexte, nous supposons qu'il y aura une accumulation des corrections traduisant l'incapacité du système à apprendre correctement la sélection avec ce contexte actif. Chaque correction fera augmenter progressivement un terme de vigilance associé au contexte actif. Lors d'une correction du comportement par le professeur, si le terme de vigilance associé au contexte actif est devenu assez élevé, alors une nouvelle cellule de lieu pourra être recrutée. Cette nouvelle cellule de lieu modifiera les entrées sensorielles impliquant un nouveau contexte à cet endroit qui permettra, éventuellement, de discriminer les situations d'apprentissage. Dans ces deux cas, l'apprentissage de l'action associée à la cellule de lieu recrutée sera fait à la fin de la correction, lorsque l'orientation du robot sera devenue correcte.

Dans l'expérience suivante, nous nous focaliserons sur la problématique de la navigation dans le cas où le robot doit réaliser des parcours différents en fonction de l'objet saisi. La tâche du robot est de choisir la bonne trajectoire et donc le bon comportement en fonction de l'objet manipulé ou devant être manipulé. Le robot navigue dans un environnement délimité. Cette navigation est apprise par interaction avec un humain qui vient guider le robot vers les endroits intéressants : lieu où le robot pourra récupérer un objet, lieux où il pourra déposer les objets. En fonction de l'objet récupéré, le robot devra aller déposer l'objet tenu à l'un des endroits montrés par l'humain. Nous décrirons dans l'article qui suit le modèle permettant la détection

des couples sensorimoteurs à inhiber et l'apprentissage des contextes multimodaux pour adapter le comportement selon l'action du professeur humain. De plus, si l'inhibition des couples déjà existants ne suffit pas à apprendre le comportement désiré, le modèle est capable de recruter d'autres couples lieu/action. Nous présenterons les résultats obtenus dans une tâche de navigation contextuelle simple, réalisée en simulation.

4.2.1 Article 4 : Apprentissage de contextes multimodaux pour l'inhibition de couples sensorimoteurs afin de moduler des comportements de navigation

de Rengervé, A., Hanoune, S., Andry, P., Quoy, M., and Gaussier, P. (2013c). Building Specific Contexts for On-Line Learning of Dynamical Tasks through Non-verbal Interaction. In *2013 IEEE International Conference on Development and Learning (ICDL) - Epigenetics and Robotics (Epirob)*, pages 1–6, Osaka, Japan

Building Specific Contexts for On-line Learning of Dynamical Tasks through Non-verbal Interaction

Antoine de Rengervé, Souheil Hanoune, Pierre Andry, Mathias Quoy and Philippe Gaussier
ETIS, CNRS ENSEA University of Cergy-Pontoise, F-95000 Cergy-Pontoise, France
{rengerve,souheil.hanoune,andry,quoy,gaussier}@ensea.fr

***Abstract*—Trajectories can be encoded as attraction basin resulting from recruited associations between visually based localization and orientations to follow (low level behaviors). Navigation to different places according to some other multimodal information needs a particular learning. We propose a minimal model explaining such a behavior adaptation from non-verbal interaction with a teacher. Specific contexts can be recruited to prevent the behaviors to activate in cases the interaction showed they were inadequate. Still, the model is compatible with the recruitment of new low level behaviors. The tests done in simulation show the capabilities of the architecture, the limitations regarding the generalization and the learning speed. We also discuss the possible evolutions towards more bio-inspired models.**

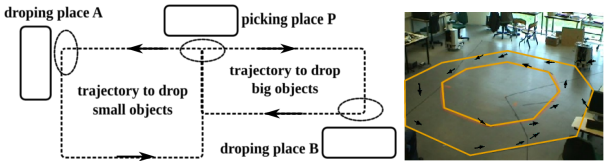


Fig. 1. *Left* Example of contextual task in navigation. A solution is to encode contexts biasing the selection of actions in the particular situations (locations (circles) + objects status) when there is a choice to make. In place P, the decision is between picking objects, moving left and moving right. In place A and B, the possible actions are dropping held object and moving to picking place. *Right* Example of typical trajectory learned through interaction as an attraction basin emerging from place-cell/orientation couples (black arrows).

I. Introduction

Action selection is the choosing of the most appropriate action out of a set of possible candidates. The word "action" can represent notions ranging from high level abstracts ("pour water in a glass") to low level motor commands ("move arm joint with chosen speed"). We are interested in a task related to the second case (Fig. 1). A robot must navigate to different places depending on the transported object. The robot takes an object at the picking place P and goes to the correct dropping places (A or B) to release the objects depending on their sizes. The robot has to select the adequate actions (moving directions i.e. low level actions) according to the current sensory inputs. After the mid 80's, solutions to action selection problem based solely on executing the steps of a given plan to achieve a goal have been progressively abandoned for more reactive behaviors [1]. Using behavior modules that always directly monitor the environment allows the system to react faster to changes. The organization of these modules can be in parallel [2] or in hierarchy [3]. It has also been argued that partial activation of reactive modules depending on a carefully designed hierarchy with attentional switching can also provide a robust solution with an easier coordination of modules [4]. In [2], the behavior modules are encoded as a condition (a combination of sensory inputs determining when the behavior can activate), an action (what the behavior consists in) and a result (expected sensory inputs after the action is performed). At first, the condition is general and the activation of the behavior is easy. Learning increases the selectivity of the condition to match the fact that the desired result cannot be obtained by the particular action in any situation. The learning is based on the correlation between the occurring of particular sensory inputs and the occurring of the result after the action is executed. Sensory inputs are progressively integrated in the condition of a behavior to better determine when it should be active, improving the coordination between the different behaviors. Reinforcement learning is another candidate solution for the action selection problem [5]. Relying on neurobiological results, computational models of the Basal Ganglia were developed to explain the action selection by humans. The properties of dopamine neurons indicate that they could implement some kind of reinforcement learning [6]. The Basal Ganglia, organized in parallel loops (potentially coding actions) with mutual inhibition, can perform the competition necessary for the action selection [7], [8]. However, both the reinforcement learning and the correlation learning are quite slow to learn. In order to have a reactive interaction, the behavior should be adapted in a faster way.

Following a simple trajectory can be encoded as an attraction basin emerging from multiple place-cell/action associations [9]. A place is encoded by merging "what" is seen with "where" it is seen[1]. A recruited place-cell (PC) responds accordingly to the distance to the encoded place with a maximal response when at the learned spot (see [9] for details). The actions are direction of movement[2]. The robot selects its moving direction depending on its place in the environment (see an example of learned trajectory in Fig. 1). The obtained navigation controller is robust due to the generalization capabilities of the place-cell recognition. New place-cell/action couples are learned from on-line non-verbal interaction [10]. When the robot moves too far from the desired trajectory, the robot is shown how to get back to the desired trajectory by forcing its orientation[3]. This corrected orientation is associated to a new learned place-cell completing the encoded attraction basin. In order to learn the task of Fig. 1, the information about

[1]The codes use visual, proprioceptive (camera orientation) and magnetic compass information to build place-cells.

[2]Orientations of movement are given with respect to an absolute reference (North). The orientation of the robot is read from a magnetic compass.

[3]In the experiments of this article, a joystick was used considering that it simulates the action of a leash.

the object (e.g. size, but it could be visual, tactile,...) should be included in the condition of the actions i.e. transforming place-cells into multimodal categories. Two approaches are possible to build the adequate multimodal categories. Local solutions to action selection can be learned and progressively adapted to enable good generalization. For instance, when the behavior of the robot is corrected, multimodal categories can be recruited and associated with the correct actions. The generalization depends on how the different modalities contribute to the context activation. Without any particular a priori, the recruited categories would include all the modalities equally. Thus, the generalization properties should be improved by learning how much each modality should contribute, with a possible pruning of the irrelevant links. The alternate solution is to start from a general category (i.e. taking only a few modalities into account) and to progressively increase the selectivity of the category (like with conditions in Maes' model [2]). Selectivity can be adapted on the basis of the feedback provided by the interaction. In the framework of the place-cell/action based controller, we propose that the place-cells be non-specific categories progressively refined by inhibiting them in situations when they predict undesired behaviors according to the interaction from the teacher. Some multimodal contexts are thus recruited to store which place-cells should be inhibited and when. A task may not be solved by only refining the existing PC/action couples. The recruitment of new PC/action couples may occur to enrich the basis of behaviors of the system. We will also show that the whole process can run in parallel with the aforementioned trajectory learning.

In Section II, we present the neural network based architecture recruiting contexts for the inhibition of irrelevant PC/action couples. The evaluation of the actions and the context recruitment are detailed with the conditions to recruit new place-cells. The model is implemented in the neural network simulator *Promethe* [11], and tested in a robotic simulation based on Webots (Cyberbotics) (Sec. III). The first experiment studies the behavior of the learning process when the contexts are not needed to learn a trajectory. Then, the model is validated in the task described in Fig. 1. The model manages to learn the task. Even though this learning scheme shows limited generalization properties, we discuss in Sec. IV whether they could be a basis for longer learning that will focus on summarizing the contexts into chunks. Such a fast adaptation of the action selection may be useful to complete and even train a slower learning network extracting the statistics of the task.

II. Model for specific context building based on interactive learning

The combination of several PC/action couples is sufficient to shape an attraction basin so that the robot follows a desired trajectory (Fig. 1 and 2, [9]). The action evaluation block in the architecture learns the contexts in which some place-cells should be inhibited. The place-cells are thus biased by the output I of the action evaluation process before the competition between the biased place-cells. The winning biased place-cell PC^I determines the selected action i.e. the orientation to be followed. Figure 3 details the action evaluation process performed in the block shown in Fig. 1. Indeed, an action is encoded as the dynamics which maintains a particular orientation of movement. During an interaction, the teacher

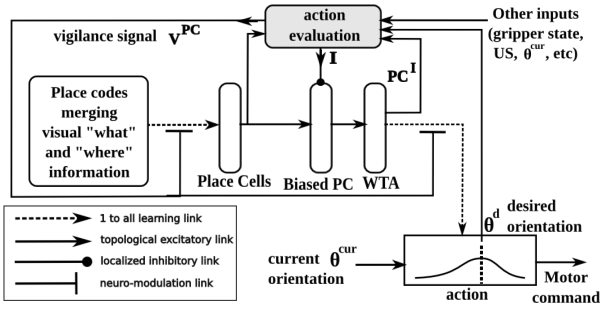


Fig. 2. Architecture for place-cell based navigation with the action evaluation part of the architecture (see Fig. 3). Sensory and predicted signals are provided to the action evaluation system. It outputs a vigilance signal and an inhibitory signal preventing some place-cells to exhibit their associated orientation. The vigilance signal can trigger the learning of a new place-cell.

corrects the behavior of the robot by opposing to the predicted dynamics i.e. by moving the current orientation away from the current desired orientation. The interaction may even be more like making the robot turn left or right than explicitly giving the exact direction to follow (no explicit supervision). Selected PC/action couples predicting rotations opposite to the executed rotation during the interaction are to be inhibited. Contexts are created to encode which place-cells should be inhibited and when. After this learning, if the behavior happens to be corrected again, the task has probably changed. Therefore, when a new interaction phase starts in an already known context, the previous associations with place-cells to inhibit will be removed to learn the new situation. However, a particular case may be taught by the teacher by alternating between correcting behaviors and evaluating the result. Though, it will be sensed by the robot as separated interaction phases. The aforementioned process is completed by a short term memory of the wrong PC/action couples to ensure the consistency of the teaching. Even if, the contextual associations are reset prematurely, the short term memory keeps the results of recent detections. Finally, repeated corrections in the same context mean the condition refining of existing behaviors fails. Hence, a new PC/action couple is recruited adding a new behavior to solve the task, possibly through condition refining.

A. Wrong action detection

The neural layer D^W outputs the result of the detection. The neurons in D^W match the existing place-cells. An activity of 1 in D^W means that the corresponding place-cell should be inhibited because the associated action is evaluated as wrong. A competition between the different biased place-cells PC^I determines the action to be performed (Fig. 2). Currently, only the action predicted by the winning biased place-cell $i_M = \underset{i}{argmax}(PC^I_i)$ can be evaluated. Therefore no wrong action detection can succeed with other PC/action couples ($D^W_{i \neq i_M} = 0$). In the case of the i^{th}_M couple, the action is evaluated as wrong if, during the interaction, the robot orientation θ^{cur} moves away from the robot proposed orientation θ^d. The teacher's control being primary, he can prevent the robot from following the desired orientation θ^d. The sensorimotor error E_r is based on the difference between these two orientations $E_r = min(\mid \theta^d - \theta^{cur} \mid, (\theta_{max} - \theta_{min}) - \mid \theta^d - \theta^{cur} \mid)$ with

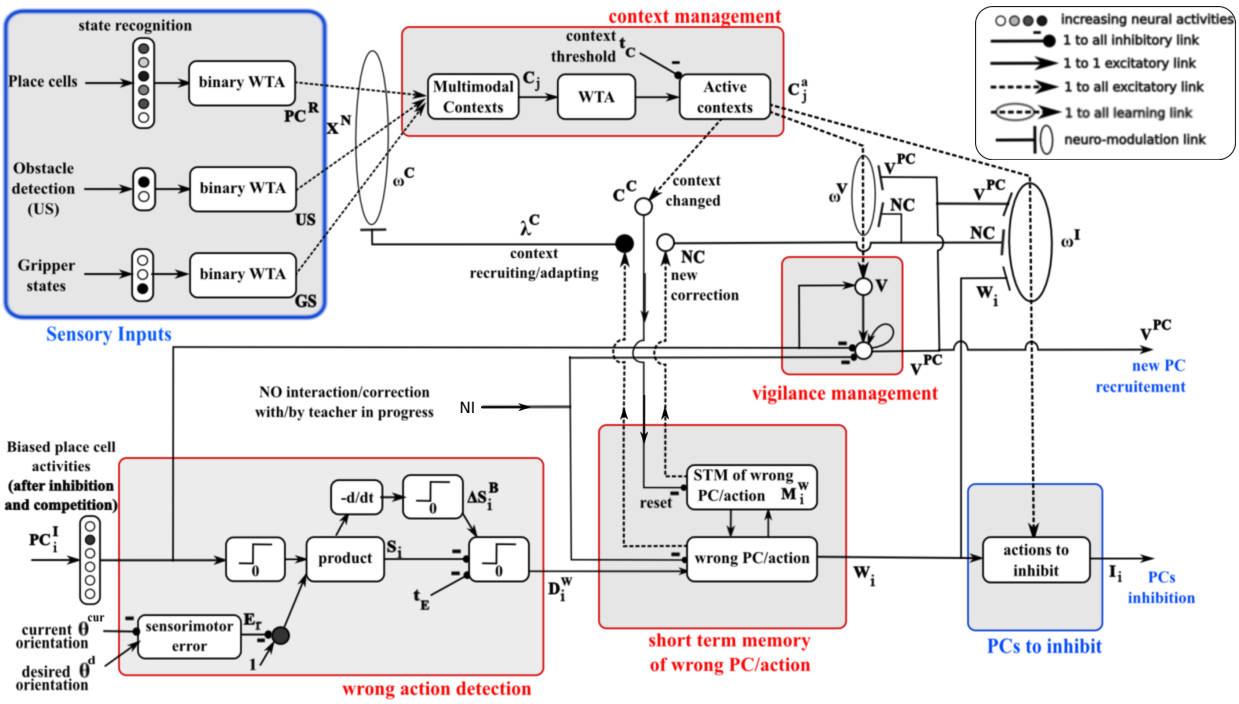


Fig. 3. Context learning to inhibit PCs associated to wrong actions and to learn new place-cells through on-line interactive teaching. Given a correction, the variation of the similarity is used to estimate if the selected action is wrong. A multimodal context is then recruited if none already encodes the situation. Each time a new interaction phase starts, the actions to inhibit are reset and the contextual vigilance increases. The short term memory of wrong actions ensures the consistency of a split interaction. Eventually, if the activity of the winning place-cell after inhibition PC^I is under the vigilance V, a new place-cell/action couple will be recruited. This architecture details the content of the action evaluation block shown in Fig. 2.

$\theta_{max} = 2\pi - \epsilon$ and $\theta_{min} = 0$. This particular computation is needed because the orientation space loops on itself (2π rad is the same as 0 rad). The similarity measure S for the i_M^{th} PC/action couple uses this error to recognize how much the current orientation is similar to the desired orientation (Eq 1).

$$S_{i_M} = 1 - \frac{2E_r}{(\theta_{max} - \theta_{min})} \quad (1)$$

with $\mathcal{H}_0$ the Heaviside function that verifies $\mathcal{H}_0(0) = 0$. The coefficient multiplying E_r normalizes the dynamics of the similarity measure (values between 0 and 1). Detecting that the $i_M{}^{th}$ PC/action couple is wrong ($D^W_{i_M} = 1$) depends on the dynamics of the similarity ($\Delta S^B_{i_M}$) as well as on its value S_{i_M} (Eq 3). An action is estimated as wrong if the evolution of the similarity indicates that the followed orientation moves away from the desired one (Eq 2) and if the similarity is low enough. When a negative variation of the similarity for the $i_M{}^{th}$ action is detected, $\Delta S^B_{i_M}$ is equal to 1. In that case, the similarity measure is compared with a similarity threshold equal to $1 - t_E$, with t_E equivalent to an error threshold.

$$\Delta S^B_{i_M}(t) = \mathcal{H}_0(S_{i_M}(t - \Delta t) - S_{i_M}(t)) \quad (2)$$

$$D^W_{i_M} = \mathcal{H}_0(\Delta S^B_{i_M} - t_E - S_{i_M}) \text{ and } D^W_{i \neq i_M} = 0 \quad (3)$$

B. Context management

Contexts are recruited when the behavior of the robot is corrected. The input X of the multimodal contexts C is the concatenation of the different discrete binary codes for each sensory modality. The raw place-cell activities PC^R give localization information. The obstacle detection US based on ultrason sensor is categorized into 2 neurons (frontal obstacle present or not). The gripper state GS is encoded by three neurons each one corresponding to an opening width. The input X is normalized and then connected to the context layer C (Eq 4). The context learning (Eq 5) is based on Adaptive Resonance Theory [12]. The maximal activity in the context layer C is compared with a vigilance threshold λ^C. If the maximum is lower, then a new context (with index r) is recruited so that the weights ω^C_{rl} reproduce the input pattern. As the input is normalized, the activity of a context is maximal when the same encoded pattern is presented again.

$$X^N_k = \frac{X_k}{\| X \|} \text{ with } X = [PC^R; US; GS] \quad (4)$$

$$\begin{cases} C_j = \sum_k \omega^C_{jk} \cdot X^N_k \\ \Delta\omega^C_{rk} = (X^N_k - \omega^C_{rk}) \text{ if } \lambda^C > \max_j(C_j) \end{cases} \quad (5)$$

with λ^C the vigilance threshold equal to 0.99 if there is an active neuron in the wrong action layer W and 0 otherwise. The active context layer C^a only contains one active neuron corresponding to the maximally recognized context $j^M = argmax_j(C_j)$ in C. The activity of the winning context must be over the context threshold t_C (Eq 6).

$$C^a_{j^M} = \mathcal{H}_0(C_{j^M} - t_C) \text{ and } C^a_{j \neq j^M} = 0 \quad (6)$$

C. Short term memory of wrong PC/action couples

During an interactive teaching phase, the PC corresponding to detected wrong actions are memorized in a short term memory. The wrong PC/action layer W depends on the new detected wrong action D_i^W as well as on the content of the memory of the recent incorrect actions M^W (Eq 7). If there is no ongoing interaction, the content of the wrong action layer is inhibited. Otherwise, any active neuron in this layer will trigger the learning of a multimodal context C and the association between this context and the neurons corresponding to the same PC in the inhibition layer I.

$$W_i = \mathcal{H}_0(M_i^W + D_i^W - 2 \cdot NI) \tag{7}$$

The short term memory M^W is fed by the wrong action layer W. The temporal forgetting γ ensures that the memory is reset within a few seconds. It can also be reset when the index of the winning context in C^a changes ($C^C = 1$).

$$M_i^W(t) = [M_i^W(t - \Delta t) - \gamma]^+ + W_i - 2 \cdot C^C \tag{8}$$

with $[x]^+ = x$ if $0 < x < 1$, 0 if $x < 0$ and 1 otherwise. A new correction phase starts whenever one neuron in the wrong actions memory becomes active. During the first iteration a new correction phase is detected ($NC = 1$). The phase ends when all neural activities decay to 0 with the forgetting or when a reset signal is received because the context changed ($C^C = 1$).

D. Learning PCs to inhibit

The inhibition layer I contains the PCs to inhibit. The associations are learned with a Hebbian like rule (Eq 9) and stored on the synaptic weights ω^I connecting the active context layer C^a with the layer I. When a new correction starts ($NC = 1$), the previously stored inhibitions are reset. When the teacher is correcting the robot, the wrong PC/actions stored in the short term memory M^W transit through the wrong PC/action layer W. The PCs to inhibit are associated with the active context in I. The context/PC associations are also reset when a new place-cell is recruited ($\alpha^{PC} = 1$).

$$\begin{cases} I_i = [W_i + \sum_j \omega_{ij}^I \cdot C_j^a]^+ \\ \Delta w_{ij}^I = W_i \cdot (C_j^a \cdot I_i - \omega_{kj}^I) - (NC + V^{PC}) \cdot C_j^a \cdot \omega_{ij}^I \end{cases} \tag{9}$$

where $V^{PC} = 1$ when a new place-cell is recruited. The PCs represented in the layer I are inhibited so that the selected orientation is predicted by one of the correct place-cells (see Figure 2).

E. Learning new PC/action couples

The aforementioned process may fail because PC/action couples with sufficiently recognized PCs may not be already encoded. The vigilance threshold V^{PC} controlling the learning of new PC/action couples is thus managed (Eq 10). Each time a new correction occurs in a context ($NC = 1$), the vigilance V associated with this context increases.

$$\begin{cases} V = [\sum_j \omega_j^V C_j^a]^+ \\ \Delta\omega_j^V = NC(PC_{i_M}^I - V) \cdot C_j^a - V^{PC} \cdot V \cdot C_j^a \end{cases} \tag{10}$$

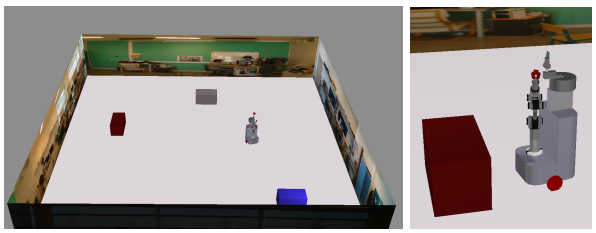


Fig. 4. Simulated environment and robot. The robot has to learn to navigate between the different blocks according to the object size signal.

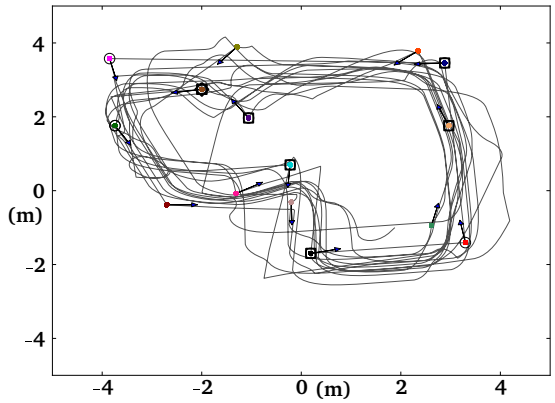


Fig. 5. Trajectory (gray line) during learning and reproduction of the task (several times consecutively). The colored dots are the learned place-cells. The chunks associated with the place-cells are represented around the corresponding dots (circle: the context exists but does not inhibit an action, square: the context does not inhibit the winning PC, star: the context is used). The arrow starting from the dots indicates the orientations that are associated to the place-cells. At the end of the learning, only one chunk still modifies the behavior. The others only inhibit PC that could not win.

with $i_M = \underset{i}{argmax}(PC_i^I)$. The activity of the winning action PC^I is compared to this vigilance threshold $V^{PC} = \mathcal{H}_0(V - PC_{i_M}^I)$ to trigger the recruitment of a new place-cell. When a new place-cell is recruited, the context/PC association for the context in which occurred the recruitment is reset ($V^{PC} = 1$ in Eq 10). The context/action association in the inhibition layer I is also reset for this context ($V^{PC} = 1$ in Eq 9). The system can switch between the PC inhibition and learning new PC/action couples.

III. Emergent attraction basins and contexts

In this section, the model is tested in two different experiments performed with a simulated robot, running with the Promethe simulator under the Webots (Cyberbotics) 3D environment. The robot is composed of a mobile platform with a camera mounted on a pan servomotor (Fig. 4). The walls of the room are covered with pictures to provide textured images for visual processing i.e. the recognition of the place-cells. In the first experiment, the robot has to learn a simple trajectory that does not require multimodal contexts. As the new model extends the model from [10], the initial test aims at studying whether the recruited contexts are useful and whether the system will rely on these contexts to tackle the task. In Figure 5, the learned place-cell/orientation associations are

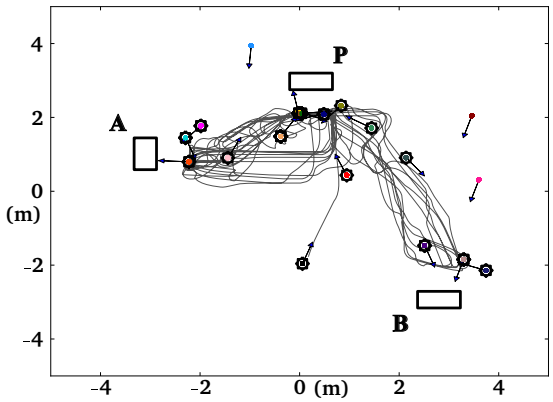


Fig. 6. Trajectory (gray line) during reproduction of the task (several times consecutively). The colored dots are the learned place-cells. The chunks associated with the place-cells are represented around the corresponding dots. The arrows starting from the dots indicate the orientations that are associated to the place-cells. Due to the effect of the chunks, the orientation may not be followed as the PC selecting this orientation is inhibited.

displayed with the followed trajectories. During the learning phase, the first time a correction is performed in a given place-cell. A context is recruited to encode the sensory configuration. Each time the orientation of the robot is corrected, the active context is associated with the inhibition of the incorrect PC/orientation couples. When a new place-cell is recruited in a context, previous associations between this context and the PCs to inhibit are unlearned. At the end of the learning phase, i.e. when the robot is finally capable of performing rounds without correction, the contexts do not influence the behavior anymore because they do not inhibit the already winning PC/orientation couple (see Fig. 5). The contexts appear as only temporarily used. Only one context still influences the behavior. It corresponds to the situation where the robot detects a wall nearby, and thus should aim at a different direction than the one associated with the winning place-cell.

In the second experiment, we focus on testing the learning of a task that requires selecting the trajectory depending on sensory information not directly related to navigation. The information is the width opening of the gripper i.e. the size of the held object. The environment contains three obstacles at interesting locations (see Fig. 4). The white one is where the robot has to take objects, the two others (red and blue) are where the robot can drop them. The colors of the obstacles are not directly used by the system. The task of the robot is to take objects at the picking place and then, depending on the simulated size of the object, move to one location or the other to place it there. The objects are simulated through the gripper state coding the object size. A correct orientation maintained during a few seconds is a prerequisite to validate the reaching to one of the places. In this experiment, there is no obstacle avoidance behavior so that the robot can face an obstacle without avoiding it. Yet, the forward speed of the robot is decreased when the robot gets closer to an obstacle so that it will not bump into it. The robot is corrected by simply changing its orientation (using a joystick) whenever it moves in the wrong direction. The trajectory followed by the robot during the reproduction of the task is presented in Figure 6. The learning phase is quite long as each new place-cell introduces a new potential context preventing previous generalization from other context and also a new possible PC/orientation that may have to be inhibited when in other contexts. The robot can reproduce the task without any correction after about fifteen rounds of the pick-and-place task using a small object and eleven rounds using a big object. As the robot only needs correction when it makes mistakes, less and less corrections are necessary as the learning process goes on.

At the end of the learning process, the robot performs the task without any correction. The learned contexts enable the robot to exhibit the correct behavior. Depending on the gripper information, the robot follows different trajectories. Even if the learned contexts are very specific and encode particular sensory configurations, the effective attraction basins can be interpreted as depending on the gripper state. The resulting attraction basins are displayed in Figure 7. Because the robot did not start from any location in the environment, the attraction basin is only correct in a limited area. Depending on its starting position, the robot can end up stuck in front of a wall (as there is currently no obstacle avoidance). The robot may also select the wrong dropping place. For instance in Fig. 7b), the dropping place on the left is where the robot should go with a small object. However, the dropping place on the bottom-right of the figure is also an attractor where the robot can go depending on its starting position. Without the corresponding learning, the robot has generalized the dropping place on the bottom-right as valid for any size of object. This is due to the fact that the learned PC/orientation couples in the vicinity of this place converge to the dropping place. The interest of the contexts appears when no object is held, by reducing the attraction basin so the robot can move to the picking place. Hence, the robot exhibits the expected contextual behavior that can emerge from the recruited contexts and the learned inhibition of PCs.

IV. DISCUSSION

In this paper we presented a model of action selection based on contexts guiding sensory-motor associations. This approach allows us to extend the place-cell/action model [9] to solve the action selection problem. In [10], the authors showed the influence of the teacher on the learned trajectory by comparing different methods of learning. The authors showed that a compromise between proscriptive and prescriptive learning was more robust and that the choices of the human interacting teacher corresponded to such compromise. We expect the same kind of influence to be at work in the contextual navigation task. The issue will be explicitly addressed in future work, in order to validate that the convergence of the context learning is not (too) dependent on the expertise of the teacher. Besides, in our experiment, we tested the model in a simulated environment, without variations and always in the same situation.

The goal of this paper was to study a minimal model that could explain the fast adaptation of multimodal behaviors during interaction with a teacher. The results showed that the proposed principles are efficient but need to be improved for a real autonomous development. First, the contexts are computed with a threshold that prevents generalization on different place-cells. Second, we also used binary values to build the sensory inputs of the contexts. This allowed us to ease the study but

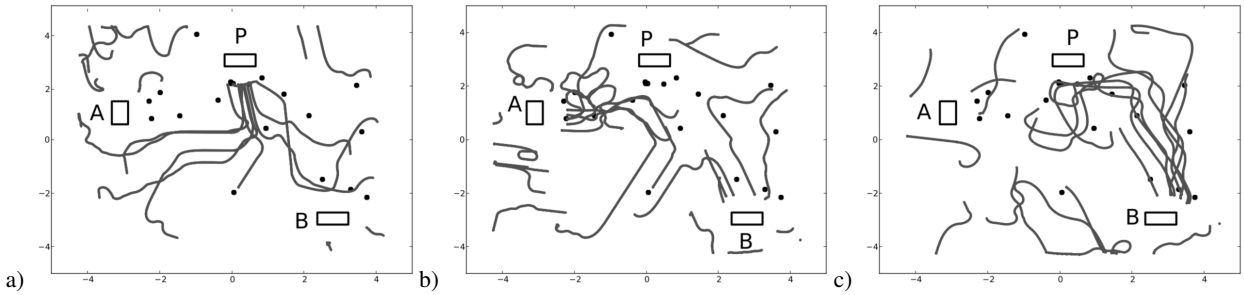


Fig. 7. Change in behavior according to the held object. Trajectories (gray) from different starting points when the gripper is holding (a) no object, (b) small object, (c) big object. Contextual attraction basins toward the pickup place P, the dropping place A and the dropping place B can emerge even though the contexts encode specific sensory configurations.

this constraint should be removed for better generalization. The generalization properties of the existing low level behaviors also determine the quality of overall behavior. There should be cooperation between the proposed fast learning of contexts and a slower learning solution improving the low level behaviors at the basis of the system.

Many neurobiological studies were dedicated to the action selection process and thus can give hints to improve the capabilities of a robotic system. Different cerebral regions are involved in the action selection process. In particular, the implication of the cortico-baso-thalamo-cortical loop was clearly exposed [13]. The GPR model [7] is based on a dynamical system approach and the internal connectivity of the BG is unknown. The CBTC model [8] tends to improve the exposure of the internal connectivity and introduces a fusion with reinforcement learning mechanisms to improve the GPR model. Those models tend to be very complex. In this paper, we focused on a minimal solution to obtain an action selection behavior, by using a selective inhibition of PC/action couples depending on the recruitment of multimodal contexts. However, this fast on-line learning is to be completed by a slower learning that could encode chunk like categories directly selecting the action to be done. The theory of chunking was first introduced in the 1950s by DeGroot [14] and Miller [15]. The main idea is that a chunk collects pieces of information in order to obtain a higher level of information coding. In a previous work [16], we suggested that a modified version of Schmajuk's and DiCarlo's learning of conditioning [17] could model the cortico-basal loop with associative conditioning in the cerebellum and resulting in the learning of chunks. Our goal is now to combine the fast on-line learning of contexts presented in this paper with the aforementioned slower learning of chunks to improve the action selection capabilities of the robot.

ACKNOWLEDGMENT

This work was supported by the AUTO-EVAL project, the INTERACT french project ANR-09-CORD-014, and the NEUROBOT project ANR-BLAN-SIMI2-LS-100617-13-01.

REFERENCES

[1] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. 2, no. 1, pp. 14–23, 1986.

[2] P. Maes and R. Brooks, "Learning to Coordinate Behaviors," in *AAAI Proceedings*, 1990, pp. 796–802.

[3] T. Tyrrell, "Computational mechanisms for action selection," Ph.D. dissertation, University of Edinburgh., 1993.

[4] J. J. Bryson, "Hierarchy and Sequence vs. Full Parallelism in Action Selection," in *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, J. A. Meyer, A. Berthoz, D. Floreano, H. Roitblat, and S. W. Wilson, Eds. Cambridge, MA: MIT Press, 2000, pp. 147–156.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 1998.

[6] W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, no. 5306, pp. 1593–1599, 1997.

[7] K. Gurney, T. J. Prescott, and P. Redgrave, "A computational model of action selection in the basal ganglia. II. Analysis and simulation of behaviour." *Biol Cybern*, vol. 84, no. 6, pp. 411–423, Jun. 2001.

[8] B. Girard, D. Filliat, J.-A. Meyer, A. Berthoz, and A. Guillot, "Integration of navigation and action selection functionalities in a computational model of cortico-basal-ganglia-thalamo-cortical loops," *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, vol. 13, no. 2, pp. 115–130, Jun. 2005.

[9] C. Giovannangeli, P. Gaussier, and G. Désilles, "Robust mapless outdoor vision-based navigation," in *IEEE/RSJ International Conference on Intelligent Robots and systems*. Beijing, China: IEEE, 2006.

[10] C. Giovannangeli and P. Gaussier, "Interactive teaching for vision-based mobile robots: A sensory-motor approach," *IEEE Trans. Syst., Man, Cybern. A*, vol. 40, no. 1, pp. 13–28, 2010.

[11] M. Lagarde, P. Andry, and P. Gaussier, "Distributed Real Time Neural Networks In Interactive Complex Systems," in *proceedings of the IEEE International Conference on Soft Computing as Transdisciplinary Science and Technology (CSTST 08)*, 2008, pp. 95–10.

[12] G. A. Carpenter and S. Grossberg, "Adaptive resonance theory (ART)," in *The handbook of brain theory and neural networks*. Cambridge, MA, USA: MIT Press, 2002, pp. 79–82.

[13] T. J. Prescott, J. J. Bryson, and A. K. Seth, "Introduction. modelling natural action selection," *Philosophical Transactions of the Royal Society B - Biological Sciences*, vol. 362, no. 1485, pp. 1521–1529, 2007.

[14] A. D. De Groot, *Thought and Choice in Chess*. Mouton De Gruyter, 2nd edition (June 1978), 1978.

[15] G. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956.

[16] S. Hanoune, M. Quoy, and P. Gaussier, "An architecture for online chunk learning and planning in complex navigation and manipulation tasks," in *IEEE International Conference on Development and Learning (ICDL) and on Epigenetic Robotics (Epirob)*, 2012, pp. 1–6.

[17] N. A. Schmajuk and J. J. DiCarlo, "Stimulus configuration, classical conditioning, and hippocampal function," *Psychological Review*, vol. 99, no. 2, p. 268–305, apr 1992, PMID: 1594726.

4.3 Sélection de comportements associés à des contextes multimodaux pour l'apprentissage d'une tâche de navigation et de contrôle d'un bras robotique

Dans la section 4.2, des contextes multimodaux sont appris pour mémoriser quand inhiber les couples sensorimoteurs qui s'opposent aux mouvements désirés par un professeur humain. Les contextes multimodaux peuvent aussi être associés directement, en positif, aux comportements pour déterminer lequel sélectionner.

Dans le cadre d'une collaboration avec le laboratoire LASA de l'EPFL [1], nous avons travaillé sur la mise en place d'une architecture permettant la réalisation de tâches impliquant le contrôle du bras d'un robot mobile. Le contrôleur pour la navigation est un contrôleur PerAc (cf. Section 1.4). Le contrôle du bras, mis en place par le LASA, s'appuie sur l'apprentissage d'un Modèle de Mixtures de Gaussiennes (GMM) pour encoder les mouvements du bras à réaliser et sur une technique de Régression sur les Mixtures de Gaussiennes (GMR) pour extraire les commandes motrices (cf. Sec. 1.1.3, et plus particulièrement l'implémentation décrite dans [Calinon et al., 2009]). Le laboratoire LASA nous a donc apporté son expertise sur l'apprentissage par démonstration avec des bras manipulateurs [Calinon et al., 2009]. Leur modèle permet d'apprendre par démonstration des séquences de gestes, en gérant la variabilité des démonstrations faites par un humain. De plus, cette variabilité peut être utilisée pour estimer la contrainte imposée sur la trajectoire à reproduire [Calinon et al., 2009]. A terme, notre objectif final est d'utiliser nos propres modèles de contrôle de bras robotique et d'y intégrer ces propriétés intéressantes. En effet, nos architectures, basées sur l'approche PerAc, nous permettent de bénéficier de propriétés dynamiques absentes de l'approche GMM/GMR. Cet objectif dépasse cependant le cadre de ce chapitre et de cette thèse. Dans cette section, nous nous focaliserons sur la question de la sélection de comportements entre navigation et contrôle d'un bras robotique. Nous décrirons une architecture réactive simple qui apprend quels comportements sélectionner. Après avoir décrit les spécificités des contrôleurs gérant la navigation et les mouvements du bras dans ce modèle intégré, nous présenterons la partie de l'architecture qui réalise l'apprentissage et la sélectionner des comportements. Nous montrerons que grâce à un apprentissage d'associations entre des contextes multimodaux et les comportements possibles, et à un mécanisme de compétition sur les comportements, le robot peut reproduire les séquences d'actions proposées par un professeur humain.

4.3.1 Comportements de navigation

Afin de réaliser la tâche souhaitée (Fig 4.1), le robot doit contrôler son orientation en fonction de sa localisation et de l'objet tenu (ici, la taille de l'objet, Fig. 4.6). Le robot utilise le retour lié à l'angle d'ouverture de la pince quand l'objet est serré. Lorsqu'une cellule de lieu PC (Place-Cell) est recrutée, elle est associée avec l'état d'ouverture de la pince G (Gripper) (catégorisée en 4 états : ouvert, fermeture faible, moyenne et complète) conduisant à la fusion des deux informations (localisation et proprioception) dans des états "cellules de lieu-pince" (PGC, Place-Gripper Cells). L'information d'ouverture de la pince permet de discriminer les différents objets suivant leur taille. L'apprentissage, en un coup lors du recrutement ($\epsilon = 1$

1. au cours du projet Européen Feelix Growing.

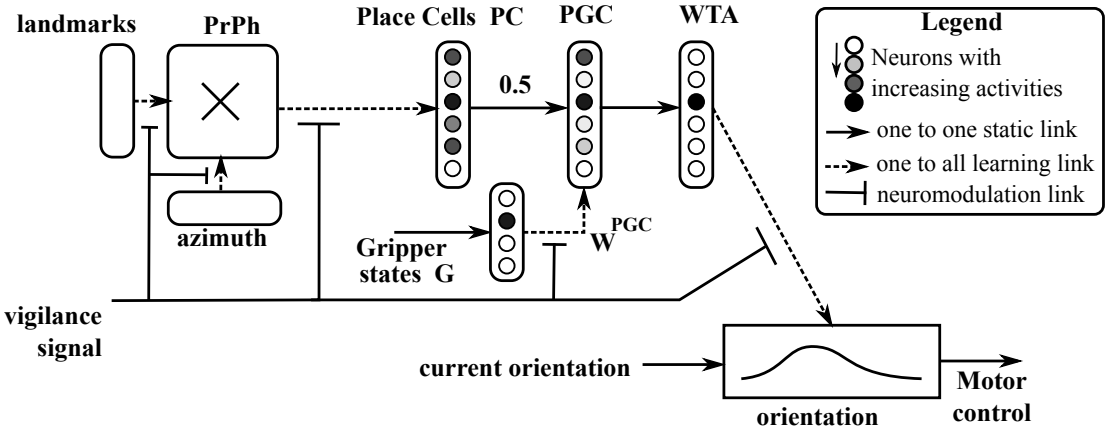


FIGURE 4.6 – Architecture pour la navigation prenant en compte l'information de l'ouverture de la pince pour construire des états lieu-pince PGC (place-gripper cells). Le signal de vigilance déclenche le recrutement d'une cellule de lieu PC et son association avec un état G décrivant l'ouverture de la pince. La fusion de ces informations donne les états lieu-pince (PGC). Ces états PGC sont associés avec la direction de déplacement désirée. La trajectoire suivie dépend ainsi de la taille de l'objet tenu. Il n'y a pas de réflexe d'évitement d'obstacles car le robot doit pouvoir rester devant le support où récupérer un objet sans chercher à s'éloigner.

lors du recrutement d'une cellule de lieu PC), suit une règle d'apprentissage associatif de type Hebbienne (4.1).

$$\begin{cases} PGC_i = \frac{1}{2}(PC_i + \sum_j W_{ij}^{PGC} G_j) \\ \Delta W_{ij}^{PGC} = \epsilon \cdot \delta_{ii_m} \cdot (PC_i \cdot P_j^G - w_{ij}^{PGC}) \text{ avec } i_m = \underset{i}{argmax}(PC_i) \end{cases} \quad (4.1)$$

Une compétition sur les états lieu-pince PGC (place-gripper cells) permet ensuite de sélectionner la cellule de lieu-pince correspondant à la situation actuelle (selon la taille de l'objet). Le gagnant de la compétition est alors associé à la direction à suivre par conditionnement sensorimoteur comme présentée dans la Section 1.4. Le robot suit directement l'orientation prédite par le modèle car, comme dans l'expérience de la section 4.2, il n'y a pas de mécanisme automatique de changement de l'orientation pour éviter les obstacles. Quand notre robot détecte un obstacle, sa vitesse diminue en fonction de la distance à l'obstacle. Ainsi, notre robot s'arrête avant de les percuter. Le fait que les orientations ne sont pas modifiées automatiquement simplifie le contrôle lorsque le robot récupère ou dépose des objets. En effet, dans les expérience, les objets sont posés sur des supports perçus comme des obstacles par le robot. Le robot devrait pouvoir apprendre dans quelles situations les obstacles peuvent être imiter automatiquement. A cette étape de notre étude, ce n'est pas encore le cas. Par ailleurs, dans les expériences réalisées (Fig. 4.6), l'apprentissage des cellules de lieu est déclenché par un signal externe arbitrairement donné par le professeur humain (l'expérimentateur appuie sur un bouton du joystick de contrôle lorsqu'il désire qu'un lieu soit appris). Mais, il est aussi possible d'utiliser d'autres types d'information comme la détection d'un changement de direction non prédit (par exemple forcé par l'action d'une laisse) pour déclencher l'apprentissage, comme cela est fait dans d'autres travaux de l'équipe [Giovannangeli and Gaussier, 2010].

Du fait des pondérations et de l'apprentissage, les cellules lieu-pince PGC suivent une

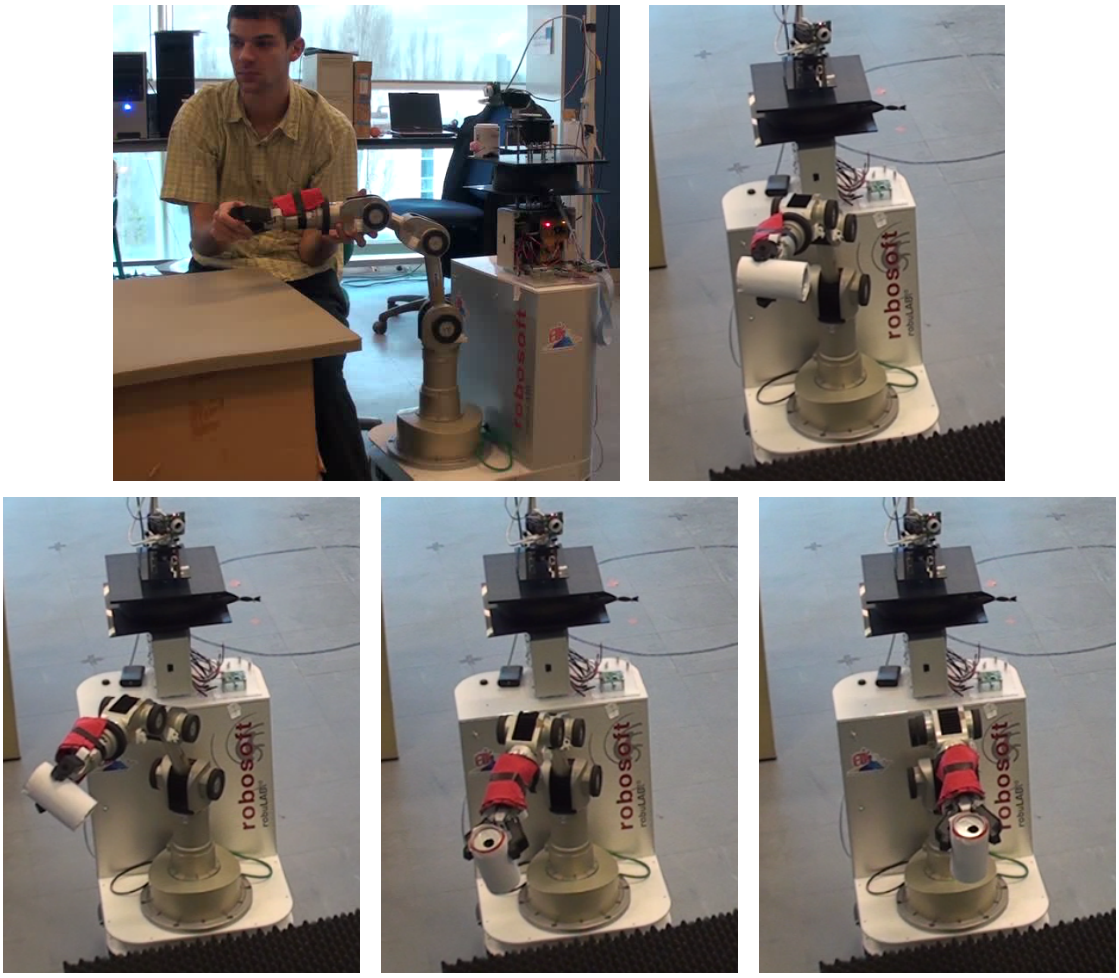


FIGURE 4.7 – Apprentissage d'une séquence de gestes par démonstration et un exemple de geste réalisé au cours de l'expérience. Les gestes sont montrés en manipulant passivement le bras. Le geste présenté ici (geste 2 dans la figure 4.10) consiste à ramener le bras dans l'axe depuis le côté droit. Dans le même temps, l'extrémité descend légèrement et le poignet effectue une rotation de 90 degrés.

hiérarchie définie par le fait que la reconnaissance de l'état de la pince prime sur la reconnaissance des cellules de lieu. Le choix des pondérations et des paramètres de la règle d'apprentissage (4.1) implique que lorsque l'association avec l'information de l'état de la pince est apprise, la connexion correspondante W_{Ij}^{PGC} aura un poids fixé supérieur à l'excitation pouvant être fournie par les cellules de lieu PC. Étant donné que les états de la pince P^G sont discrétisés et binaires, la contribution de l'état de la pince l'emportera forcément sur la contribution des activités des cellules de lieu.

4.3.2 Contrôle du bras robotique

Le contrôleur du bras robotique est basé sur l'algorithme GMM/GMR décrit à la Section 1.1.3. Trois séquences de mouvements sont apprises chacune à partir de 3 démonstrations (Fig. 4.7). Chaque séquence a été montrée au système séparément et préalablement aux expéri-

ences en utilisant l'approche décrite dans [Calinon et al., 2009]. Ce contrôleur communique avec le reste de l'architecture qui est implémentée sous forme neuronale. Pour le reste de l'architecture, les traitements réalisés par le contrôleur GMM/GMR sont transparents. En effet, les entrées et les sorties de ce contrôleur sont considérées par le reste de l'architecture comme des activations neuronales. Les entrées du contrôleur GMM/GMR sont la proprioception du bras et un contexte donnant quel mouvement reproduire, et les sorties sont les commandes en vitesse pour le bras.

4.3.3 Apprentissage et sélection des comportements actifs suivant des contextes multimodaux associés

Nous présentons ici le metacontrôleur qui permet de sélectionner le comportement désiré (Figure 4.8). Dans l'architecture actuelle, le rôle du metacontrôleur est de sélectionner quel

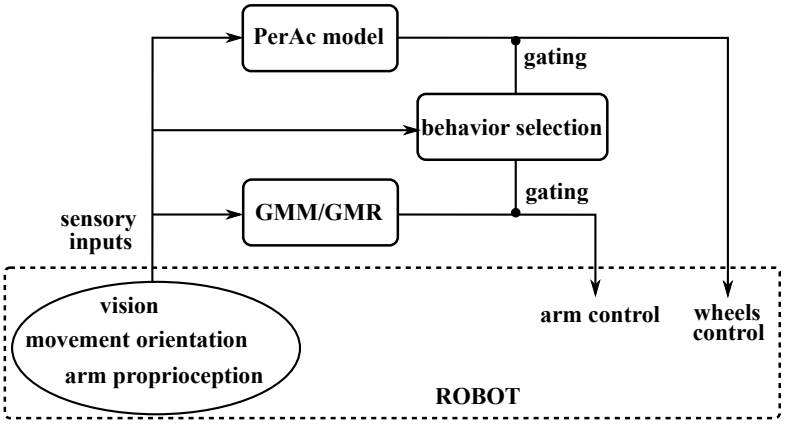


FIGURE 4.8 – Architecture pour le contrôle du robot durant l'expérience de tri avec déplacement.

module ou routine peut être activé parmi les six possibilités qui sont : naviguer, ouvrir la pince, attendre la présence d'un objet à emporter, et déclencher l'un des trois gestes appris. Les trois gestes appris sont : un mouvement depuis une position dans l'axe du robot (position de repos) vers une position plus haute et sur sa droite avec rotation du poignet, le mouvement de retour à la position de repos (fig. 4.7) et un mouvement circulaire depuis la position de repos. Le choix du comportement actif dépend du contexte du robot correspondant à son état sensoriel. Un contexte est défini comme étant la conjonction des différentes entrées sensorielles (senseurs ultrason détectant la proximité des obstacles, angle d'ouverture de la pince, état du bras après le dernier geste réalisé, localisation du robot) catégorisées séparément (information binaires) (Fig. 4.9). Ainsi, chaque neurone de contexte C_i, où i est l'indice du neurone, correspond à un unique état pour le robot. Afin de simplifier, dans la version du modèle implémentée, tous les contextes possibles [2] sont préexistants mais ils auraient pu être recrutés sur la base d'un changement sensoriel détecté avec les mêmes résultats. Un modèle est en cours de test dans le cadre de la thèse de M. Souheil Hanoune au sein de notre laboratoire. Un système de recrutement permettrait cependant de réduire le nombre de contextes nécessaires.

2. i.e. résultant des différentes combinaisons d'entrées sensorielles catégorisées et binaires

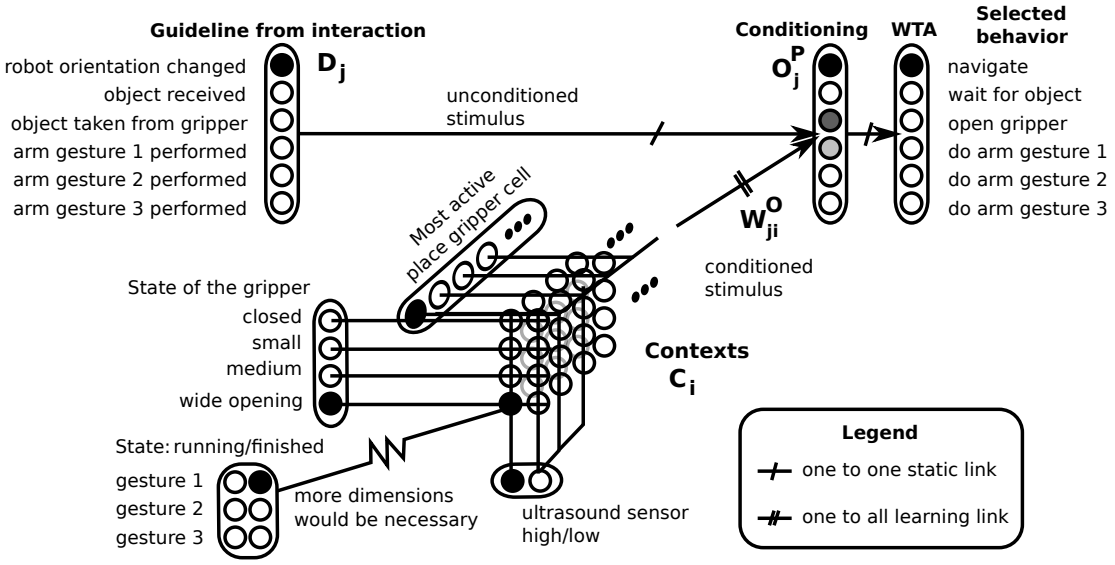


FIGURE 4.9 – Schéma du réseau de neurones utilisé pour la sélection de l'action basée sur des contextes. Les contextes sont la conjonction des différentes entrées sensorielles catégorisées et binaires. Dans la version implémentée du modèle, tous les contextes possibles sont préexistants mais ils auraient pu être recrutés. Les instructions données par l'interaction sont estimées à partir d'entrées sensorielles comme le déplacement du joystick et les senseurs sur la pince (pour forcer l'ouverture ou la préhension). Un conditionnement classique réalise l'apprentissage de l'association du comportement à réaliser avec le contexte courant.

A partir du contexte actif, le robot doit sélectionner l'action à réaliser. En réponse aux actions du robot, l'humain peut agir pour que le comportement du robot soit celui désiré en toute situation. L'humain produit ainsi des directives non verbales implicites que le robot utilise pour apprendre la tâche. Par exemple, l'humain va retirer un objet tenu par le robot afin de le déposer. Le robot détecte alors un changement sur les entrées sensorielles (proprioception, toucher) correspondant à l'enlèvement de l'objet, accompagné par une ouverture réflexe de la pince. Le robot pourra associer le comportement permettant d'obtenir le même changement sensoriel (e.g. routine d'ouverture de la pince) avec le contexte courant. Dans cette version initiale, les six modules de comportement sont pré-cablés sur la base de variations particulières des entrées sensorielles : usage du joystick pour forcer à naviguer, donner un objet dans la pince pour attendre de récupérer un objet, retirer l'objet tenu pour ouvrir la pince et trois boutons différents pour initier l'un des trois gestes de manipulation avec le bras. L'apprentissage est basé sur un conditionnement simple minimisant l'erreur entre la sortie inconditionnelle et la sortie conditionnelle au sens des moindres carrées [Widrow and Hoff, 1960]. L'intervention de l'humain génère des variations perceptives catégorisées a priori et définissant les stimuli inconditionnels D_j. Dans notre modèle, l'entrée inconditionnelle est égale à la sortie inconditionnelle. Ainsi, la sortie prédite O_j^P pour le j^e neurone de la couche du conditionnement converge vers la valeur D_j (4.2). Quand le neurone de contexte C_i activé durant l'apprentissage est à nouveau actif, le système prédit alors le comportement qui a été associé.

$$O_j^P = \sum_i W_{ji}^O \cdot C_i \quad \text{où} \quad \Delta W_{ji}^O = \mu(t) \cdot (D_j - O_j^P) \cdot C_i \tag{4.2}$$

Le stimulus inconditionnel D_j vient aussi moduler le facteur d'apprentissage $\mu(t)$ pour que le conditionnement ne soit réalisé que lorsqu'il y a interaction. L'apprentissage vient modifier les poids des connexions W_{ji}^{O} entre les contextes multimodaux C_i et les sorties prédites O_j^P. Une compétition sur les sorties prédites par le conditionnement détermine le comportement qui doit être activé. Seules les commandes motrices générées par le module actif sont traitées. Ainsi, si le metacontrôleur ne prédit aucune activation de modules alors tous les comportements sont inactifs et le robot reste en attente.

4.3.4 Expérience et résultats

Dans l'expérience réalisée, le robot est dans une salle et doit réaliser des actions alternant déplacement entre différents lieux (éloignés de deux à trois mètres les uns des autres) et manipulation d'un objet en fonction de la taille de l'objet. La Figure 4.10 décrit l'environnement

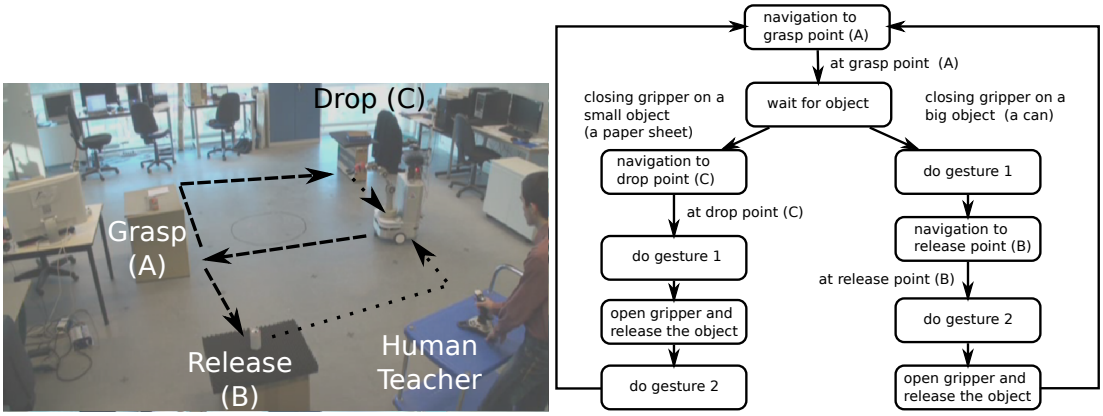


FIGURE 4.10 – Vue d'ensemble de la tâche. Au point de récupération, le robot reçoit un objet et choisit en fonction de la taille de l'objet reçu de quel côté il se rend ensuite. L'auto-localisation du robot est basée sur la vision permettant une navigation robuste. Le bras est contrôlé par la solution avec Mixture de Gaussiennes (GMM/GMR) permettant de reproduire les mouvements montrés précédemment. Ces gestes seront réalisés avec ou sans canette tenue. A droite : automate de la séquence d'actions devant être apprise par le robot.

dans lequel cette tâche a été réalisé. Il consiste en une seule salle dans laquelle différents lieux particuliers sont matérialisés par des obstacles pour le robot (lieu A,B,C dans les Figures 4.10 et 4.11). Le lieu A est le lieu de récupération des objets. Le robot doit attendre qu'on lui donne un objet. Quand il l'a récupéré, il doit alors décider où se rendre. Dans notre cas, la direction à suivre dépend de la taille de l'objet tenu (large pour une canette, étroite pour une enveloppe) et le robot devra donc naviguer vers deux endroits différents. Les deux lieux où déposer des objets sont notés B et C sur la figure. Dans cette expérience, nous avons supposé que le robot n'avait pas besoin d'états internes. En effet, les informations sensorielles sont suffisantes pour permettre de discriminer les différentes situations et sélectionner le bon comportement à tout instant.

Le comportement a été enseigné au robot avec succès via un apprentissage en ligne et grâce aux démonstrations montrant dans quels endroits attraper et déposer des objets, en fonction de leur taille. L'expérience réalisée a duré environ 1h. La figure 4.11 correspond à la trajectoire du robot durant cette expérience. L'apprentissage a duré 22 min soit 2 passages pour la boucle avec les canettes et 1 passage pour la boucle avec les enveloppes. Le robot reproduisait ensuite les

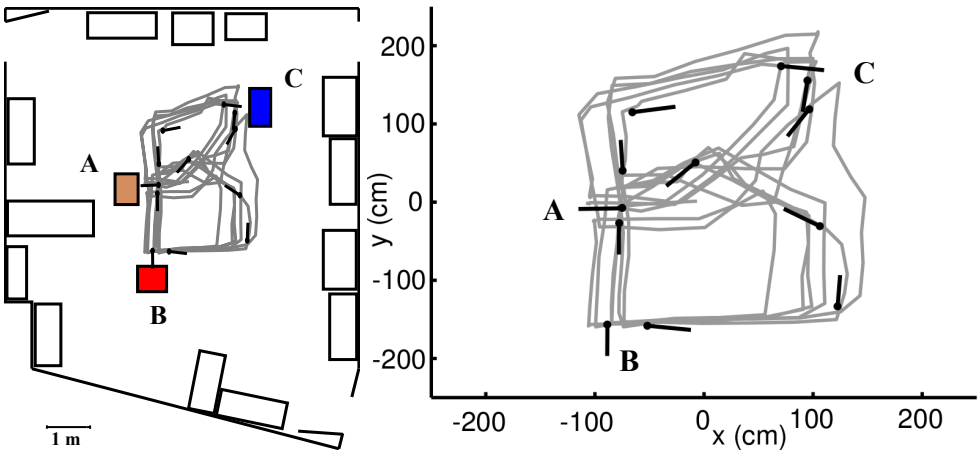


FIGURE 4.11 – Trajectoire et bassin d'attraction appris durant la tâche de tri avec navigation. A gauche le plan de la salle et à droite un zoom sur la trajectoire et les couples lieu-action appris (points et segments noirs). Les couples lieu-action ne définissent pas une cuvette qui garantirait un bon comportement quelque soit le point de départ. Grâce à la présence des obstacles fixes, la trajectoire ne dévie pas.

séquences de manière autonome en fonction de l'objet qu'on lui donnait. Au total, le robot a réalisé le trajet pour déposer des canettes 6 fois et celui pour déposer des enveloppes 5 fois.

Dans cet expérience, les couples lieu-action appris (figure 4.11) ne définissent pas un bassin d'attraction qui assurerait le bon comportement de navigation quel que soit le point de départ. En effet, il manque plusieurs couples lieu/action pour que le bassin d'attraction soit correctement généralisé. La présence des obstacles fixés dans l'environnement a permis de maintenir la même trajectoire au cours de l'expérience. Le robot ne s'est donc pas retrouvé dans de nouvelles situations lui permettant de compléter l'apprentissage du bassin d'attraction. L'un des rôles du professeur doit être de confronter le robot à des situations en dehors du comportement normal, par exemple en montrant au robot comment revenir sur la trajectoire désirée depuis des lieux hors du chemin habituel.

Dans sa version actuelle, le modèle, bien que suffisant pour résoudre la tâche proposée, est limité en terme de capacité de généralisation. En particulier, considérer un nouveau contexte à chaque nouvelle situation empêche toute capacité d'anticipation par généralisation. De plus, dans ces situations, le robot se met en attente de la démonstration de l'humain pour savoir quoi faire. Les contextes devraient être recrutés quand c'est nécessaire, plutôt qu'à chaque changement perceptif détecté, afin de laisser la possibilité au système de généraliser. L'apprentissage de ces contextes devrait alors aboutir à une généralisation correcte de la sélection de comportements sur les situations nouvelles.

4.4 Discussion

Dans ce chapitre, nous avons abordé la problématique de la gestion de comportements dans le cas d'une tâche intégrant des comportements de navigation et de contrôle d'un bras.

Dans la section 4.1, nous avons montré que des comportement simples exécutés en parallèle par un robot mobile portant un bras robotique pouvait permettre de réaliser un comportement en apparence sophistiqué. La mise en parallèle d'un contrôle visuomoteur pour le bras robotique et

d'un contrôleur de l'orientation du robot mobile a permis d'obtenir que le robot se déplace pour aller attraper les objets d'intérêt loin de lui, sans synchronisation ni planification.
Nous avons remarqué ensuite que même si l'exécution de plusieurs comportements en même temps peut être intéressant, il faut aussi pouvoir inhiber certains comportements quand c'est nécessaire. Nous avons alors proposé d'évaluer les comportements pertinents à travers les couples sensorimoteurs les définissant. L'évaluation est faite en situation d'interaction en comparant les changements de proprioception attendus par les couples sensorimoteurs et les changements de proprioception produits par les actions du professeurs humain corrigeant le comportement du robot. Les couples sensorimoteurs qui s'opposent aux actions correctrices du professeur sont associées avec des contextes multimodaux recrutés. Lorsque la situation de la correction se représente, les couples associés au contexte reconnu sont inhibés. Dans l'expérience réalisée (Sec. 4.2), ces mécanismes ont permis à notre robot d'apprendre à suivre différentes trajectoires vers différents lieux en fonction d'une information de contexte (e.g. la taille de l'objet transporté s'il y en a un).
Les contextes multimodaux peuvent aussi être associés positivement avec les comportements réalisés. Dans la section 4.3, cet apprentissage et une compétition sur les comportements activés ont permis à un robot mobile équipé d'un bras robotique de gérer différents contrôleurs (PerAc pour la navigation, GMM/GMR pour le bras robotique) et de réaliser les séquences d'actions correspondant à une tâche de tri et de manipulation d'objets avec déplacement entre différents lieux.

Dans ces différentes expériences nécessitant d'apprendre à sélectionner des comportements, nous nous sommes focalisé sur l'apprentissage rapide en ligne de la tâche grâce à l'interaction avec un professeur humain. La qualité de l'apprentissage en interaction dépend de la capacité du robot à apprendre en ligne durant l'interaction et à confronter au fur et à mesure les prédictions du modèle appris avec la supervision de l'humain. De ce point de vue, le modèle décrit dans la section 4.2 semble le plus adapté. En s'appuyant sur la dynamique de la correction, le robot peut modifier rapidement son comportement en même temps que l'humain modifie son orientation. Ainsi, le robot peut prédire en anticipation l'orientation que veut lui donner le professeur. Cela permet d'alléger la démonstration puisqu'il n'est plus nécessaire de montrer exactement le comportement désiré. Il suffit de guider assez le robot pour qu'il puisse estimer le comportement attendu.

Dans les expériences des sections 4.2 et 4.3, le robot ne disposait pas d'un mécanisme d'évitement d'obstacles modifiant automatiquement son orientation. Ainsi, le robot conservait son orientation vers un obstacle quand il devait récupérer un objet dessus. Cependant, un tel mécanisme d'évitement d'obstacles serait nécessaire pour avoir une navigation autonome robuste. L'objectif serait d'avoir un robot qui puisse naviguer entre différents lieux en évitant les obstacles, et qui soit aussi capable d'apprendre à s'arrêter face à un obstacle pour récupérer un objet ou le déposer dessus. Pour cela nous proposons d'appliquer le principe d'apprentissage de l'inhibition des couples sensorimoteurs (Sec. 4.2) à l'évitement d'obstacle. Un comportement d'évitement d'obstacles génère une dynamique motrice visant à diminuer l'activation des capteurs détectant les obstacles. Si la correction appliquée par le professeur humain fait augmenter cette activation, i.e. oriente le robot vers l'obstacle, alors l'évitement d'obstacles devrait être inhibé [3]. Cette expérience serait aussi l'occasion d'étudier la transition entre inhibition de

3. *Nota Bene :* quand le robot se rapproche de l'obstacle, la vitesse d'avancement linéaire du robot diminuerait pour éviter de le percuter.

configurations particulières (obstacle détecté selon un certain angle) et l'inhibition d'un comportement dans son ensemble. En effet, une généralisation des couples sensorimoteurs vers le comportement devrait être possible. Ainsi, si plusieurs actions proposées par un comportement se révélaient toutes inadaptées, le système pourrait inférer que ce comportement n'est pas approprié et cesser de le prendre en compte pour déterminer les actions futures.

Les travaux futurs devront aussi porter sur la réalisation de la tâche de tri avec déplacements sans utiliser le contrôleur GMM/GMR. A la place, les séquences de gestes motivées apprises sous la forme de cartes cognitives (chapitre 3) et le contrôleur visuomoteur (chapitre 2) déjà utilisé à la section 4.1 pourront être utilisés. Ces deux modes de contrôle feront partie des comportements pouvant être sélectionnés pour réaliser la tâche. Dans ce chapitre, nous avons présenté deux manières d'apprendre quel comportement sélectionner. Ces deux solutions s'appuient sur des contextes multimodaux mais diffèrent quant à l'utilisation de ces contextes. Dans la section 4.2, ils servent à inhiber les comportements indésirables tandis que dans la section 4.3, ils viennent exciter les comportements désirés. Cette différence d'utilisation implique des mécanismes de sélection différents : dans le premier cas, les comportements devraient pouvoir s'exécuter en même temps s'ils ne sont pas inhibés, tandis que dans le second cas, une compétition stricte sélectionne un unique comportement vainqueur. Compte tenu de l'avantage d'avoir plusieurs comportements actifs en même temps, il sera préférable dans une version évoluée du modèle d'avoir une compétition souple voire une combinaison des comportements pondérés par les contextes multimodaux actifs. Un tel procédé rappelle l'approche Yuragi [Fukuyori et al., 2008] que nous avons présenté pour le contrôle moteur dans le chapitre 2. Dans la section 4.2, un retour fourni par l'action du professeur humain permet d'évaluer le comportement choisi au niveau des couples sensorimoteurs. En fonction de ce retour, l'attracteur guidant le comportement (couples sensorimoteurs) pourrait être maintenu ou inhibé afin de permettre une exploration des autres attracteurs existants jusqu'à exécuter un comportement correspondant au retour donné par l'humain.

L'avantage d'inhiber les comportements existants est d'avoir une utilisation peu invasive des contextes puisqu'on laisse les comportements non inhibés s'exprimer normalement. De plus, il est envisageable que les nouveaux contextes appris ne soient utilisés que temporairement. En effet, le comportement récurrent pourrait compléter son apprentissage en recrutant de nouveaux couples sensorimoteurs rendant inutiles certain contextes appris. Par exemple, dans la figure 5 de l'article 4 (sec. 4.2.1), une ronde classique est apprise en interaction pendant que des contextes multimodaux sont recrutés. A la fin de l'apprentissage de la ronde, les contextes inhibent seulement des couples sensorimoteurs dont la reconnaissance - sans l'inhibition - était de toute façon inférieure à celle du couple sélectionné. Ces contextes sont inutiles et pourraient être oubliés. En revanche, les contextes multimodaux utilisés pour exciter des comportements ou des actions permettent d'affiner le comportement du robot en associant certaines actions ou comportements à des conjonctions particulières de modalités (localisation, proprioception,etc.).
Cette différence de gestion est certainement liée aussi à une différence en terme de constante de temps pour l'apprentissage. Les contextes pour l'inhibition peuvent être appris rapidement et temporairement. En revanche, les contextes multimodaux excitateurs devraient être appris plus lentement afin d'aboutir à une sélection plus fiable et plus robuste des comportements grâce à l'accumulation d'expériences. L'hypothèse d'une future combinaison de ces deux solutions complémentaires nous amène à la problématique de l'apprentissage développemental de la sélection d'actions.

L'un des éléments clés pour améliorer la sélection d'actions serait d'apprendre des contextes multimodaux permettant des prédictions dans de nouvelles situations et suffisamment précis pour permettre de discriminer correctement ces situations. Cet apprentissage des bonnes catégories multimodales nous amène à la théorie des chunks. Cette théorie a été développée dès les années 50 par Miller [Miller, 1956] et DeGroot [De Groot, 1978]. L'idée principale est qu'un chunk rassemble des éléments d'information pour obtenir un codage de l'information compressé et de plus haut niveau. La thèse de Souheil Hanoune, actuellement en cours au laboratoire ETIS, porte sur ce type d'apprentissage. Dans [Hanoune et al., 2012], les auteurs ont proposé une version modifiée de l'apprentissage par conditionnement de Schmajuk et DiCarlo [Schmajuk and DiCarlo, 1992] qui modélise la boucle cortico-basale et un apprentissage conditionnel avec le cervelet pour obtenir l'apprentissage de chunks. Cet apprentissage, validé en simulation, nécessite la présentation des situations apprises à de multiples reprises et prend donc du temps. Les travaux futurs dans cette direction devront confronter cet apprentissage à des situations réelles complexes comme la tâche présentée dans ce chapitre. Durant l'apprentissage des chunks, le robot devra rester capable de modifier rapidement son comportement selon l'interaction avec un professeur humain. Il serait donc intéressant de combiner l'apprentissage en ligne et rapide décrit dans ce chapitre avec un apprentissage plus lent de chunks permettant d'obtenir les contextes permettant une bonne généralisation de la sélection des comportements.

Publications personnelles

D'halluin, F., de Rengervé, A., Lagarde, M., Gaussier, P., Billard, A., and Andry, P. (2010). A state-action neural network supervising navigation and manipulation behaviors for complex task reproduction. In *Proceedings of the tenth international conference on Epigenetic Robotics*, pages 165–166, Orenas Slott, Sweden

Communications dans des ateliers de travail internationaux

de Rengervé, A., Hirel, J., Quoy, M., Andry, P., and Gaussier, P. (2011). A simple neural network controller merging different behaviors for collector robots. In *International Workshop on bio-inspired robots*, Nantes, France

J'ai des questions à toutes vos réponses.

– Woody Allen

CHAPITRE 5

Conclusion et perspectives

L'objectif de cette thèse était de montrer que l'apprentissage de tâches de natures différentes peut être abordé comme un problème d'apprentissage d'attracteurs sensorimoteurs à partir d'un petit nombre de structures non spécifiques à une tâche donnée. Nous avons donc proposé une architecture qui permet l'apprentissage et l'encodage d'attracteurs pour réaliser aussi bien des tâches de navigation que de contrôle d'un bras.

Nous avons développé dans cette thèse le modèle Dynamic-Muscle PerAc (DM-PerAc) pour le contrôle d'un bras robotique. Il prolonge notre approche sensorimotrice (PerAc, [Gaussier and Zrehen, 1995]) et hérite donc de bonnes propriétés de généralisation pour l'apprentissage de comportements visuomoteurs. Ainsi, dans notre modèle, la dynamique des couplages sensation-action permet de générer des comportements robustes et fiables. Ces couplages définissent des bassins d'attraction dynamiques dont la construction (apprentissage) est très éloignée du contrôle optimal mais qui, in fine, partagent un certain nombre de propriétés similaires à celles obtenues avec des algorithmes de contrôle optimal. Par exemple en navigation (chapitre 1), suivant les contraintes apprises durant l'interaction avec un humain, l'orientation du robot n'est changée que lorsque le robot risque de sortir du bassin d'attraction appris. Ce mécanisme s'apparente au principe du minimum d'intervention [Todorov and Jordan, 2003] qui serait appliqué au maintien du robot dans un bassin d'attraction comportemental. D'autre part, des tests de comparaison entre les solutions PerAc et LWPR (Locally Weighted Projection Regression, [Vijayakumar et al., 2005]), et PerAc et GMM/GMR (Gaussian Mixture Model/Gaussian Mixture Regression, [Calinon et al., 2009]) ont montré que les performances d'apprentissage étaient comparables dans les cas considérés. Notre modèle semble plus efficace pour un apprentissage de tâches rapide et en ligne avec une interaction directe avec un humain. Il présente aussi l'avantage d'être plausible biologiquement et de permettre d'expliquer la construction des comportements en faisant le lien entre les associations multimodales apprises au niveau neuronal, les attracteurs sensorimoteurs (mouvements, points fixes, trajectoires) et les comportements (imitation, apprentissage de tâche). Nous avons insisté sur l'importance des interactions entre les différentes structures impliquées dans le contrôle qui apportent des capacités de traitement et d'apprentissage non spécifiques à une tâche donnée.

DM-PerAc permet de combiner toutes les propriétés de l'architecture PerAc avec la souplesse d'un contrôle en impédance [Hogan, 1984a] (chapitre 2). Ce contrôle en impédance a fait ses preuves sur des problèmes d'interactions physiques et de compliance. Il nous permet aussi d'avoir un contrôle moteur proche de celui des muscles. Le modèle DM-PerAc gère des activa-

tions musculaires en modifiant les raideurs des ressorts simulant les muscles. Le contrôle obtenu correspond ainsi à un contrôle en impédance à la différence près que les schémas d'activations musculaires peuvent correspondre aussi bien à un attracteur postural, à un bassin d'attraction décrivant une trajectoire ou à un mélange des deux.

Nous avons ensuite étudié comment obtenir différents comportements d'imitation et d'apprentissage de tâches en interaction avec un partenaire (chapitre 3). L'apprentissage d'un contrôleur visuomoteur, utilisant des attracteurs posturaux, et des mécanismes simples suffisent pour obtenir un comportement d'imitation. L'apprentissage des attracteurs posturaux définit des contraintes en posture avec lesquelles un retour d'erreur sur les mouvements produits permet d'adapter les activations musculaires. Celles-ci sont alors associées à des catégories visuomotrices apprises fusionnant informations visuelles et proprioceptives. Le contrôleur ainsi construit réalise un *homéostat visuomoteur* i.e. essaye de maintenir un équilibre entre ces deux informations (chapitre 2 et 3). A l'opposé d'une démarche ingénieur, notre modèle tire avantage des capacités de perception limitée du robot. L'*ambiguïté de la perception* associée à un comportement homéostatique garantit, comme dans de précédents travaux, l'émergence de comportements d'imitation de bas niveau cognitif (ici mimer un geste par ex.).
Pour aller plus loin, nous avons montré que, grâce à ce mécanisme d'homéostasie complété par un *apprentissage de séquences visuelles* et la capacité d'*inhiber ses mouvements* durant l'observation, notre robot pouvait imiter en différé. Le robot peut ainsi apprendre par observation une séquence de gestes et la reproduire selon ses capacités visuomotrices. Nous avons validé ces principes sur l'apprentissage de séquences de gestes pour attraper et déplacer des objets suivant une démonstration visuelle.

Dans la section 1.5.2, nous avions soulevé la question de la place des neurones miroirs [Gallese et al., 1996; Rizzolatti and Craighero, 2004] dans le développement des capacités d'imitation et d'apprentissage de tâches grâce à un professeur. Dans notre architecture, les neurones miroirs ne sont pas un pré-requis pour le développement des capacités d'imitation mais plutôt un effet de bord des apprentissages associatifs réalisés. Ainsi, les neurones dans la carte visuomotrice sont activés durant l'observation et la réalisation des mouvements. Leur schéma d'activation est similaire à l'activation des neurones miroirs participant à une "résonance de bas niveau". Cependant, ces neurones ne peuvent pas expliquer la "résonance de haut niveau" des neurones miroirs. Pour cela, il est nécessaire de faire appel à la notion de but. Nous nous sommes donc intéressés à la mémorisation de plusieurs séquences et à leur sélection en fonction de contextes motivationnels.

Nous avons utilisé les principes des cartes cognitives, développées en navigation, pour permettre un apprentissage et une planification sur différentes séquences selon plusieurs contextes motivationnels. Un graphe (carte cognitive) des différentes séquences de transitions possibles est appris. Certaines transitions sont associées avec les contextes motivationnels définissant alors les "buts" à atteindre. Nous avons désigné par "but" la transition finale d'une séquence de transitions réalisées et qui est associée à l'obtention d'une récompense. La propagation d'un potentiel d'activité, des contextes motivationnels vers les transitions-buts puis vers toutes les transitions précédentes du graphe qui leur sont associées permet de mettre en place un gradient d'activité que le robot pourra remonter en sélectionnant la transition avec le potentiel maximal. Ces mécanismes permettent d'apprendre des buts liés à des contextes motivationnels et, ainsi, de réaliser des tâches dépendant d'un contexte motivationnel. Nous avons validé ce modèle sur l'apprentissage d'une tâche de prise et dépôt (tri) de canettes selon leur couleur, apprise par manipulation passive du bras du robot. Nous avons montré que les buts pouvaient être appris et modifiés

simplement grâce à une interaction sociale où l'humain signifie, par une expression faciale par exemple, que l'action du robot est correcte ou non. En particulier, nous avons proposé une extension du modèle de carte cognitive qui permet de tester les buts appris à travers leur influence sur le choix de l'action en cours. Le système est ainsi capable de déterminer quel est le neurone (but) contrôlant le comportement suivi à un instant donné. Cette détection a facilité l'adaptation du comportement du robot via une interaction sociale. Dans les expériences, un renforcement négatif exprimé par l'expérimentateur pouvait indiquer au robot que le but suivi était incorrect. Ce but étant inhibé, le robot changeait rapidement son comportement pour rejoindre un autre but accessible. L'acquisition par le robot de ces différents comportements nous a ensuite conduit à la problématique de leur sélection.

Nous avons choisi d'aborder la problématique de l'apprentissage en ligne et en interaction de la sélection de comportement dans le cadre riche (modalités sensorielles et comportements variés) d'une tâche mêlant navigation et contrôle d'un bras robotique (chapitre 4.3).
Nous avons tout d'abord remarqué et illustré dans une expérience de robotique que la synergie de comportements s'exécutant en parallèle peut suffire à réaliser des comportements relativement complexes. Un robot mobile qui s'oriente vers un objet et qui aligne visuellement sa pince sur ce même objet a été capable de se déplacer pour aller attraper un objet sans qu'aucun processus de synchronisation ou de planification ne soit nécessaire. Cependant, une telle synergie ne peut suffire à résoudre toutes les tâches. Il doit être possible de sélectionner les comportements à exécuter indépendamment.
Nous avons alors étudié comment des contextes pouvaient servir à inhiber les comportements détectés comme inappropriés ou à exciter les comportements désirés. Nous nous sommes d'abord intéressés à l'inhibition de comportements via l'inhibition des couples sensorimoteurs correspondant. Nous avons proposé un mécanisme qui évalue les couples sensorimoteurs suivant les éventuelles actions correctrices d'un professeur humain, et s'en sert pour créer des état "contextes" permettant d'inhiber ponctuellement les couples sensorimoteurs problématiques. Nous avons validé ce modèle sur l'apprentissage de différentes trajectoires dépendant d'un contexte (e.g. avoir une dynamique sensorimotrice différente en fonction du type d'objet tenu). Enfin, nous avons testé l'utilisation de contextes pour renforcer l'activation de certains comportements et biaiser leur sélection. Grâce à ce modèle, notre robot a pu apprendre à réaliser une tâche de tri avec déplacement entre différents lieux (prise de l'objet, déplacement selon celui-ci, dépôt). Nous avons ainsi mis en évidence les différents rôles que peuvent jouer les contextes multimodaux pour l'apprentissage en ligne de la sélection de comportement. Cela nous a amenés à aborder la question de l'intégration de la supervision du professeur dans l'évaluation des comportements réalisés.

Dans la suite, nous allons discuter trois grandes questions que posent les travaux que j'ai réalisés :

- Comment améliorer l'apprentissage des bassins d'attraction et affiner la cinématique des gestes ?
- Comment évaluer et sélectionner des stratégies ?
- Comment gérer les buts et les contextes dans des tâches complexes ?

5.1 Comment améliorer l'apprentissage des bassins d'attractions et des gestes du bras ?

Les architectures que nous avons développées ont permis d'apprendre le contrôle sensorimoteur des mouvements d'un bras robotique. Nous reviendrons sur la qualité des gestes générés par notre architecture y compris lorsqu'ils sont planifiés. Nous discuterons alors comment utiliser notre modèle pour expliquer une propriété des mouvements biologiques : le schéma tri-phasique d'activations musculaires. Enfin, nous décrirons un mécanisme pouvant compléter notre architecture pour améliorer le processus d'apprentissage des attracteurs sensorimoteurs.

5.1.1 Comment apprendre et gérer des couples état/activations musculaires pour construire des gestes ?

L'un des avantages majeurs de DM-PerAc est que le schéma d'activation des muscles permet de définir la posture désirée (co-activation) ou la direction de mouvement souhaitée (activation "asymétrique" i.e. l'un des muscles antagonistes est inactif). Suivant un apprentissage interactif des bassins d'attraction (PerAc), il devrait être possible de jouer sur la compliance du bras pour le "pousser" dans la direction désirée afin qu'il apprenne les mouvements à réaliser. Dans un cas classique d'apprentissage de type PerAc, les couples sensorimoteurs associent simplement l'état sensoriel S du système au moment de la correction avec une commande extraite à partir de la direction du mouvement désiré. La direction du mouvement de l'extrémité du bras robotique peut être évaluée dans l'espace visuel ou dans l'espace proprioceptif. Une simple différenciation temporelle sur le retour proprioceptif suffirait pour estimer la direction dans l'espace proprioceptif. Dans le cas de la direction du mouvement dans l'espace visuel, nous préconisons d'utiliser le flot optique afin d'avoir une estimation de la direction de mouvement fiable et robuste au bruit. Les directions de mouvement pourraient être catégorisées pour être utilisées ensuite dans l'apprentissage des couples sensorimoteurs définissant le bassin d'attraction sensorimoteur.
Cependant, les commandes motrices nécessaires pour suivre une direction de mouvement choisie dépendent aussi de la posture du bras robotique à cause du contrôle musculaire réalisé. Dans DM-PerAc, les muscles sont contrôlés par le changement des raideurs de ressorts avec une longueur de repos fixée. Ainsi, comme l'étirement du muscle diffère selon les positions, le couple généré par la contraction musculaire peut varier pour une même activation musculaire. Les mêmes activations musculaires envoyées au bras robotique peuvent générer des directions de mouvement différentes parce que la position du bras n'est pas la même. La connaissance de la direction de mouvement désirée n'est donc pas suffisante pour extraire la commande motrice produisant le mouvement attendu.
La solution est d'apprendre une base de couples associant les bonnes commandes motrices avec des états (P-DM) fusionnant les informations de Posture et de Direction de Mouvement (dans l'espace visuel ou dans l'espace proprioceptif). Nous proposons que l'apprentissage des couples associant les états P-DM et les commandes motrices suive un protocole similaire à celui mis en place dans [Srinivasa et al., 2012] pour apprendre le modèle DIRECT. Notre robot réaliserait un babillage moteur en produisant des mouvements oscillants autour de certaines postures grâce à différentes activations musculaires asymétriques. Dans le même temps, ces activations musculaires seraient associées avec les états P-DM activés par les retours sensoriels de la posture du bras et de la direction du mouvement perçue (dans l'espace visuel ou proprioceptif). Une fois cet

apprentissage réalisé, le bassin d'attraction pourrait être construit par le recrutement de couples sensorimoteurs associant l'état sensoriel S du système avec l'action (commande motrice) prédite par l'état P-DM (ou la combinaison d'états P-DM) activé par le mouvement désiré.
Cependant, l'apprentissage réalisé ne peut pas garantir que la direction résultant de cette commande motrice sera exactement celle désirée. Nous défendons que l'approche Yuragi de Fukuyori et collègues [Fukuyori et al., 2008] peut aussi être appliquée pour sélectionner et combiner les commandes motrices prédites par les états P-DM afin d'adapter les mouvements réalisés. Suivant l'approche Yuragi, la modulation des contributions des différents couples état P-DM/commande motrice par rapport à du bruit ajouté sur la commande motrice, permettrait au robot de produire des mouvements suivant des directions intermédiaires non apprises. Nous avons déjà implémenté le principe du Yuragi dans le cas d'une sélection d'attracteurs posturaux avec un retour lié à l'information visuelle (chapitre 1). L'extension à la combinaison de mouvements devrait être assez simple.

5.1.2 Comment expliquer le schéma tri-phasique d'activation musculaire et améliorer la fluidité des gestes planifiés ?

Des études chez l'humain sur les schémas d'activations musculaires observées par EMG (ElectroMyoGrammes) ont montré l'existence d'un schéma appelé tri-phasique [Sanes and Jennings, 1984; Chiovetto et al., 2010]. Lors d'un mouvement rapide, les muscles antagonistes alternent leur activation tout en maintenant au minimum leur co-activation. Dans un premier temps, un accroissement fort de l'activation du muscle agoniste initie le mouvement. Puis, le muscle antagoniste prend le relais. L'activation du muscle agoniste devient faible tandis que celle du muscle antagoniste devient forte. Durant cette phase, le mouvement décélère. Enfin, dans un troisième et dernier temps, le muscle agoniste vient compléter le mouvement par une contraction d'intensité peu élevée permettant la stabilisation sur la posture finale.
Une explication très bas niveau de ce phénomène s'appuie sur la manière dont les activations musculaires génèrent la contraction des muscles. Suivant les travaux séminaux de [Feldman, 1966], différents modèles de contrôle neuromusculaire [Zajac, 1989; Feldman and Levin, 2009] considèrent que la commande des muscles se fait via le changement des seuils de repos de muscles modélisés par des ressorts non-linéraires (les muscles ne peuvent que se contracter). La théorie du point d'équilibre développée par Feldman [Feldman and Levin, 2009], aussi appelée la théorie du contrôle du seuil, défend que le mouvement est généré par le réflexe myotatique [1] et contrôlé grâce aux seuils de repos des muscles définissant les positions d'équilibre des articulations. Le schéma d'activation tri-phasique peut alors être expliqué par le fait que, suivant la posture initiale et la posture finale, les seuils de repos des muscles impliqueront une poussée d'activité sur un seul des muscles avant la co-activation lorsque le mouvement se termine [Feldman et al., 1990]. La théorie du contrôle du seuil a néanmoins été très critiquée [Bizzi et al., 1992; Gottlieb, 2000] entre autres sur l'absence de justifications au contrôle moteur et à la présence des schémas d'activation tri-phasiques dans le cas d'une désafférentation artificielle [2] des muscles chez un sujet. La possibilité d'effectuer le contrôle dans un cadre désafférenté et d'adapter le contrôle en fonction de la stratégie (pointage vs impact [Gottlieb, 2001]) indique

1. Le réflexe myotatique désigne la contraction réflexe d'un muscle déclenchée par son propre étirement.
2. Grâce à l'injection d'une substance chimique adéquate, les retours proprioceptifs peuvent être bloqués empêchant ainsi les réflexes myotatiques (désafférentation). Le schéma d'activation tri-phasique reste cependant observable chez ces sujets.

que le contrôle moteur peut aussi réaliser un contrôle direct du couple. Afin de prendre en compte les multiples propriétés observées dans ces différentes expériences, il faudrait sans doute considérer un modèle comme [McIntyre and Bizzi, 1993] qui combinerait à la fois une boucle réflexe et un contrôle direct des activations musculaires (donc des couples moteurs) afin d'expliquer le contrôle dans le cas désafférenté. Dans le modèle de [McIntyre and Bizzi, 1993], le contrôle du torque dépend de 3 contributions : (1) une contribution axée sur un contrôle central de la trajectoire du point d'équilibre, et une contribution liée à la boucle réflexe avec (2) un signal de retour en position et (3) un signal de retour en vitesse. La contribution (1) permet un contrôle en boucle ouverte (cohérent avec la désafférentation) tandis que les deux boucles de contrôle rétroactives sont disponibles pour réguler la position et la vitesse désirées. Un tel modèle n'apporte cependant pas d'explication satisfaisante quant à l'existence du schéma tri-phasique dans le cas désafférenté.

Le schéma d'activation tri-phasique correspond à l'alternance des activations entre des muscles antagonistes au cours d'un mouvement rapide vers une position cible. L'utilisation exclusive d'attracteurs posturaux, comme c'était le cas dans nos expériences, implique que les muscles sont co-activés en permanence et ne peut donc pas expliquer l'apparition du schéma tri-phasique. Notre architecture basée sur DM-PerAc a été utilisée pour la planification de mouvements exécutés sous la forme de séquences de points de passage. La contrainte du passage était déterminée par la reconnaissance d'un état et pouvait donc être plus ou moins souple. Cependant, la dynamique des mouvements reflétait le fait qu'il s'agissait de gestes successifs passant de position en position. Dans l'idéal, la suite de gestes devraient être réalisée dans un même mouvement, de manière fluide. L'utilisation exclusive d'attracteurs posturaux appris avec DM-PerAc s'est donc révélée trop limitée pour expliquer le schéma tri-phasique d'activations musculaires et pour permettre une bonne dynamique des gestes planifiés.
Cependant, dans la section 5.1.1, nous avons proposé comment améliorer les mouvements réalisés avec un apprentissage sensorimoteurs des directions à suivre dans notre modèle DM-PerAc. Une amélioration équivalente est possible pour l'apprentissage des transitions. En effet, les transitions (visuelles et proprioceptives) codent implicitement une information de direction, ce qui justifie leur potentielle association avec des commandes motrices asymétriques. On remarquera cependant que, contrairement aux transitions entre états proprioceptifs, les transitions entre état visuels ne contiennent pas l'information de la posture motrice et donc ne seraient pas assez discriminantes. La solution serait d'utiliser des transitions entre les états visuomoteurs, existants dans DM-PerAc, afin de garder à la fois les avantages des états moteurs et ceux des transitions visuelles. Ces transitions seraient associées directement avec les activations musculaires. L'apprentissage des auto-transitions, i.e. des transitions d'un état vers lui-même, devrait aboutir à l'apprentissage d'un schéma de co-activations musculaires, tandis que les transitions entre deux états différents seraient associées à des activations "asymétriques"[3]. Durant un mouvement, le robot réaliserait les transitions entre états successifs afin de rejoindre la cible. Une fois l'objectif atteint, le robot utiliserait alors une auto-transition pour conserver la même posture. L'élan initial donnerait la direction à suivre, corrigée progressivement au fur et à mesure des transitions prédites. Les mouvements réalisés seraient donc nettement plus fluides.

L'association directe des transitions avec les activations musculaires, dans notre modèle DM-

3. On remarquera l'avantage de cet apprentissage sur les transitions apprises en navigation. En effet, en navigation, les auto-transitions sont difficiles à définir par une orientation à suivre. Ici, la dualité dans les schémas d'activation musculaire se prêterait parfaitement à l'apprentissage des transitions comme des auto-transitions.

PerAc, permettrait d'expliquer l'apparition d'un schéma tri-phasique d'activations musculaires durant la réalisation d'un mouvement. Considérons les mouvements réalisés lors d'un geste rapide avec notre modèle. Initialement, la cible étant un peu éloignée, une transition serait sélectionnée, impliquant qu'une partie des muscles serait activée pour donner l'élan dans la direction désirée. Un dépassement de la posture étant probable, le mouvement devrait alors être corrigé, potentiellement par la réalisation d'une transition ramenant le bras vers la posture désirée. Dans ce cas, les muscles jusqu'alors inactifs prendraient le relais. Il y aurait ainsi l'alternance d'activations tandis que les mouvements se stabiliseraient peu à peu sur la posture finale désirée. Le mouvement résultant serait cependant différent de celui attendu (dépassements et oscillations versus arrêt direct sur la cible). Notre hypothèse est que des capacités de prédiction des conséquences des actions permettraient d'anticiper les dépassements et de produire les corrections en avance. Le mouvement généré alors se terminerait bien sur la position cible, sans dépassement, et avec le schéma d'activation tri-phasique complet.
L'utilisation d'une prédiction des conséquences des actions est une solution relativement classique [Wolpert et al., 1998; Bhushan and Shadmehr, 1999; Todorov, 2004] pour améliorer les performances d'un contrôleur. Elle s'appuie sur une copie de la commande motrice (copie efférente) transmise à un modèle direct, souvent appris, pour prédire les conséquences perceptives. L'intégration des conséquences perceptives dans la boucle sensorimotrice aurait alors deux avantages. Une fusion des perceptions réelles et prédites permettrait de contrôler les mouvements avec une anticipation sur les résultats et ainsi d'obtenir le schéma tri-phasique. D'autre part, dans le cas désafférenté, le retour donné par le module prédisant les conséquences suffirait à générer les mouvements avec le schéma d'activation tri-phasique attendu. Un paramètre temporel devrait déterminer si la prédiction est à plus ou moins long terme. Le schéma d'activation prendrait alors des formes différentes permettant d'anticiper plus ou moins bien les corrections à réaliser. Il restera à résoudre la question de la gestion de ce paramètre temporel, en étudiant en particulier ses dépendances pour obtenir un mouvement correctement exécuté. Les modèles bio-inspirés comme [Wolpert et al., 1998; Bhushan and Shadmehr, 1999] associent généralement ces capacités de prédictions sensorimotrices au cervelet. Ce problème de modélisation du cervelet et de ses capacités de prédictions est abordé au sein de notre équipe par les travaux de thèse de David Bailly.

5.1.3 Comment accélérer et optimiser l'apprentissage des attracteurs sensorimoteurs ?

L'apprentissage des attracteurs sensorimoteurs pourrait être plus rapide et les capacités de mémoire mieux gérées grâce à un processus de correction des mouvements basé sur l'adaptation neuronale des activités musculaires. Dans le chapitre 2, lorsqu'une nouvelle posture devait être apprise, un nouvel attracteur postural était recruté. La commande motrice prédite dépendait alors de poids synaptiques initialisés à des valeurs aléatoires faibles. En conséquence, le mouvement généré après le recrutement était un mouvement faible vers une position aléatoire. Durant l'apprentissage de la posture, les mouvements étaient progressivement corrigés par la modification des poids synaptiques. Avec la méthode de correction utilisée (e.g. apprentissage par renforcement, méthode Yuragi), le temps nécessaire pour réaliser la correction dépendait de l'erreur initial et de la configuration sensorimotrice désirée. Ainsi, comme l'erreur initiale pouvait être importante, le temps nécessaire pour apprendre la nouvelle posture était généralement assez

long.
Nous proposons que le processus de correction des mouvements soit réalisé de manière neuronale, dans un processus séparé du recrutement et de l'apprentissage des attracteur qui ne seraient réalisés que lorsque la commande adéquate est trouvée. Lorsqu'une nouvelle posture doit être apprise, le processus de correction entrerait en action. La différence avec la solution utilisée dans le chapitre 2 est que le contrôle appris continuerait de prédire une commande motrice permettant de rejoindre une posture proche. Nous faisons en effet l'hypothèse que les attracteurs déjà appris prédiraient une commande motrice permettant de rejoindre une posture relativement proche de celle désirée. L'erreur devrait donc être nettement plus faible et plus rapide à corriger que précédemment. Au fur et à mesure de la correction, le système devrait parvenir à générer une commande motrice permettant de rejoindre la posture avec un niveau de précision suffisant (par rapport à un critère fixé). Un nouvel attracteur serait alors recruté et associé à la commande motrice utilisée qui fusionnerait la commande motrice prédite et la correction appliquée. D'une certaine manière, il y aurait transfert de savoir puisque les prédictions des attracteurs déjà appris seraient réutilisées dans l'apprentissage du nouvel attracteur. La durée entre le moment où l'apprentissage de la nouvelle posture est déclenché et le moment où l'attracteur est correctement appris serait ainsi réduit par rapport à la solution utilisée dans le chapitre 2. Le mécanisme proposé permettrait d'accélérer l'apprentissage de la coordination visuomotrice.
On voit apparaître un facteur important pour le contrôle : le temps mis pour corriger les mouvements sous une certaine précision (fixée). Cette durée pourrait devenir le critère pour déclencher l'apprentissage : si le temps mis pour atteindre la posture désirée est trop long, un nouvel attracteur codant cette configuration motrice devrait être appris. Utiliser ce critère pour éviter de recruter un nouvel attracteur serait avantageux. D'une part, le nombre de neurones disponibles pour gérer des attracteurs est limité. D'autre part, le recrutement de nouveaux attracteurs peut influencer les comportements déjà construits et impliquer divers apprentissages supplémentaires. Par exemple, en créant de nouveaux états, de nouvelles transitions deviennent réalisables. Des apprentissages supplémentaires sont alors nécessaires pour que les cartes cognitives puissent planifier avec ces nouvelles transitions.
Ainsi, les travaux futurs devraient étudier le choix des critères de précision et de durée de correction afin de gérer au mieux le compromis entre apprendre de nouveaux attracteurs ou prendre le temps de corriger les mouvements. Les efforts futurs sur l'apprentissage des comportements moteurs, en termes de vitesse d'apprentissage et de qualité des mouvements, devraient améliorer les comportements d'interaction qui en découlent.

5.2 Comment évaluer et sélectionner des stratégies ?

Grâce à notre architecture, notre robot peut exhiber différents comportements offrant différentes manières d'apprendre en interaction et différentes manières de réaliser une tâche. Le terme de "stratégie" permet de désigner des façons différentes de construire des comportements qui agissent dans le même espace et peuvent donc se retrouver en conflit. Le système doit alors être capable d'évaluer de manière autonome les différentes stratégies possibles afin de choisir la bonne. Nous avons évoqué dans la discussion du chapitre 3 les travaux de [Nguyen and Oudeyer, 2012], [Dollé et al., 2010] et [Jauffret et al., 2013] sur la sélection de stratégies. Dans [Nguyen and Oudeyer, 2012], les auteurs proposent une algorithme permettant à un robot d'apprendre quelle stratégie d'apprentissage utiliser en fonction de l'état du système. Les stratégies d'ap-

prentissage considérées sont une exploration autonome active[Baranes and Oudeyer, 2010], une émulation d'un professeur sélectionné activement parmi les professeurs disponibles et une reproduction mimétique des gestes d'un professeur sélectionné. La capacité à sélectionner de manière autonome la stratégie la plus appropriée permet de réduire le temps d'apprentissage et offre une solution complémentaire à notre approche. Les travaux de Dollé et collègues [Dollé et al., 2010] considèrent deux stratégies de navigation basées l'une sur des cartes cognitives et l'autre sur une stratégie stimulus-réponse basée sur un apprentissage par renforcement [Sutton and Barto, 1998]. Les auteurs utilisent une estimation de la performance de chaque stratégie grâce à un apprentissage par renforcement [Sutton and Barto, 1998] pour basculer d'une stratégie à l'autre. Dans les travaux de Jauffret et collègues [Jauffret et al., 2013], la confiance dans une stratégie est évaluée selon ses capacités de prédiction en terme de perception attendue. Cette méthode a été appliquée à la sélection entre une stratégie de suivi de route (point de fuite) et une stratégie de navigation sensorimotrice basée sur des associations lieu-action. Dans le chapitre 4, nous avons utilisé une comparaison entre la dynamique proprioceptive attendue et la dynamique forcée par le professeur humain pour inhiber des attracteurs incorrects. Cette approche complète celle de [Jauffret et al., 2013] en se focalisant sur le cas particulier de l'apprentissage en interaction avec prise en compte de la supervision de l'humain. Au lieu de considérer la perception dans son ensemble, nous nous sommes focalisés sur le retour perceptif lié à la dynamique motrice i.e. l'évaluation du fait que les mouvements du robot, forcés par le professeur, suivent ou non l'attracteur prédit. On remarquera que, dans les expériences que nous avons faites, ce sont des couples sensorimoteurs qui ont été inhibés et non des stratégies. Il s'agit d'une première étape. Nous pensons que l'évaluation des couples sensorimoteurs liés à une stratégie permettra à terme de construire la mesure de confiance dans cette stratégie. Des contextes appris peuvent alors permettre de mémoriser les situations et le résultat de l'évaluation pour accélérer la prise de décision la fois suivante. L'apprentissage de contextes pouvant se généraliser correctement est un facteur-clé pour améliorer la sélection des comportements.
Dès les années 50 par Miller [Miller, 1956] et DeGroot [De Groot, 1978] ont développé l'idée de créer des chunks rassemblant différentes sortes d'information pour obtenir un codage de l'information compressé et de plus haut niveau. Cette théorie des chunks nous semble une voie intéressante pour avoir des contextes capables de mieux généraliser. La thèse de Souheil Hanoune, actuellement en cours au laboratoire ETIS, s'intéresse à l'apprentissage de chunks pour améliorer la sélection d'action. Dans [Hanoune et al., 2012], les auteurs ont proposé une version modifiée de l'apprentissage par conditionnement de Schmajuk et DiCarlo [Schmajuk and DiCarlo, 1992] qui modélise la boucle cortico-basale et un apprentissage conditionnel avec le cervelet pour réaliser un apprentissage de chunks. Cet apprentissage, validé en simulation, nécessite la présentation des situations apprises à de multiples reprises et prend donc du temps. Les expériences que nous avons réalisées constituent "un" point de départ pour les travaux futurs qui devront tester l'apprentissage de chunks dans des situations complexes. L'utilisation de chunks pour représenter les contextes sera essentiel pour permettre l'apprentissage de tâches complexes.

5.3 Quels apprentissages et quand pour la planification de tâches complexes ?

Grâce à notre modèle, nous avons pu enseigner différentes tâches à notre robot sous la forme de séquences d'actions. La réutilisation des comportements déjà maîtrisés permettrait

d'accélérer l'apprentissage de nouvelles tâches tout au long de l'existence du robot [Thrun and Mitchell, 1995]. Cependant, notre robot est encore loin de pouvoir réaliser des tâches complexes qui impliqueraient d'exécuter différentes sous-tâches déjà maîtrisées.

Suivant la description en schèmes proposée par Piaget [Piaget, 1936; Drescher, 1991], un comportement rassemble trois aspects : une condition sensorielle qui détermine quand réaliser l'action, l'action en elle-même, et les résultats sensoriels attendus lorsque l'action est réalisée dans les conditions prévues. Un résultat sensoriel devient un but lorsqu'il existe une motivation pour l'obtenir. Dans [Drescher, 1987; Maes and Brooks, 1990], une action est évaluée en fonction des résultat sensoriels qu'elle produit dans une condition donnée. Un tel découpage de la structure d'une action n'est cependant pas nécessaire pour générer des comportements. En particulier, le résultat produit n'a pas besoin d'être clairement défini. Par exemple, dans le cas de la construction de bassins d'attraction sensorimoteurs en navigation, la cellule de lieu gagnante (condition) détermine directement l'action à réaliser, ce qui a pour résultat de changer l'orientation du robot mais surtout sa position lorsque le robot se déplace (résultat). Cette information de la position d'arrivée n'a pas besoin d'être connue a priori pour permettre un comportement de suivi de chemin par exemple.

Néanmoins, cette constatation ne semble plus tenir lorsque l'on considère la planification. Ainsi, pour mettre en place des capacités de planification, nous avons vu que la création et l'utilisation de transitions était nécessaire. En navigation, une transition lie la cellule de lieu de départ avec la cellule de lieu d'arrivée. Une transition correspond donc au fait de lier la condition de réalisation de l'action avec la conséquence de sa réalisation. Suivant le même raisonnement que celui qui a justifié l'emploi de transitions pour la navigation avec des cartes cognitives, il faudrait avoir des transitions entre des contextes multimodaux. Ces contextes décriraient les sous-tâches et permettrait de les activer permettant la planification d'une séquence de sous-tâches pour réaliser des tâches complexes. Le contexte représentant le résultat de la réalisation de la sous-tâche serait alors son but. L'encodage de ce but serait bien plus riche, en termes de modalités impliquées, qu'une simple transition-but utilisée dans notre modèle. C'est la raison pour laquelle un apprentissage de ce graphe s'appuyant uniquement sur la détection des transitions entre contextes semble vouée à l'échec. Le modèle serait nécessairement tributaires des limites liées à la reconnaissance des contextes condition/but définissant les transitions. Nous savons que l'apprentissage de bon contextes (de bon chunks) est loin d'être un problème résolu. Or, sans la reconnaissance des bons contextes, les transitions liées aux sous-tâches réalisées ne peuvent pas être détectées correctement. Une transition non détectée ne serait pas être intégrée dans le graphe, et sa construction lacunaire empêcherait de planifier correctement.

Une solution serait de revenir à une évaluation plus dynamique du comportement réalisant une sous-tâche. Considérons une sous-tâche définie par la propagation d'un gradient d'activités dans une carte cognitive. Ce contexte motivationnel à l'origine de ce gradient peut définir des buts qui sont des attracteurs comportementaux pour le système. En effet, le système, par ses actions, remonte ce gradient quel que soit son état initial. Ainsi, observer une transitions entre des contextes définis ne serait pas la seule manière de détecter qu'une sous-tâche particulière est exécutée. Une tâche pourrait être définie par le fait que les actions du robot suivent bien l'attracteur comportemental associé, détecté en s'assurant que les actions remontent bien le gradient propagé par le but correspondant à la tâche. Cela correspond au principe utilisé au chapitre 3 qui a permis de tester les buts appris pour déterminer le but poursuivi par le robot (en vue de l'inhiber lorsqu'un renforcement négatif était reçu). Dans cette solution, le système reste cepen-

dant encore tributaire de la bonne détection des transitions utilisée par la carte cognitive pour définir les séquences de mouvements. Éventuellement, il serait possible de descendre au niveau de l'évaluation dynamique des comportements selon la méthode illustrée dans le chapitre 2 lors de la comparaison des propriétés de PerAc et des DNF. Nous avions utilisé une évaluation du comportement réalisé basé sur l'intégration du produit des sensations par les actions afin de construire le champ décrivant la perception du robot selon le formalisme décrite dans [Maillard et al., 2005]. Un tel système permettrait de construire un estimateur du comportement réalisé plus fiable que la détection d'une transition entre deux états dont la robustesse de la reconnaissance dépend de multiples facteurs (nombre de modalités impliquées, bruit,...). Ce système déjà utilisé dans [Jauffret et al., 2013] pour réaliser l'évaluation des stratégies en vue de leur sélection semble donc une piste intéressante pour la mise en place de tâches complexes réutilisant des tâches déjà apprises.

5.4 Pour finir...

Dans cette thèse, nous nous sommes intéressés à plusieurs situations d'interaction en particulier : interaction en face à face, interaction côte à côte, apprentissage en passif (Figure 5.1). Notre robot a réalisé des imitations en face à face de gestes observés en direct ou précédemment

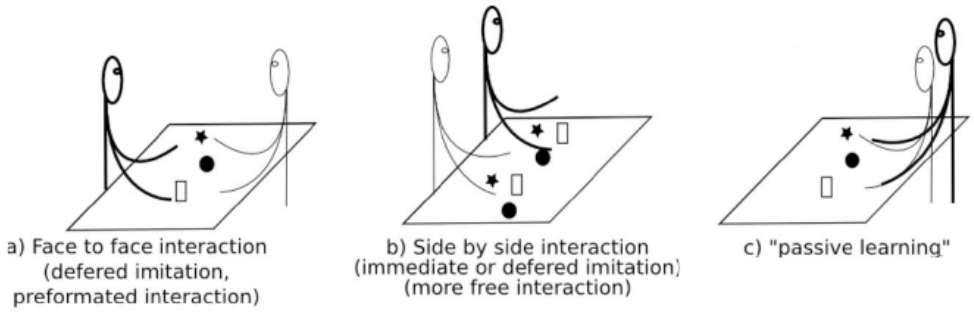


FIGURE 5.1 – Les différentes situations d'apprentissage par imitation.

montrés. Notre modèle a aussi permis que le robot apprenne des tâches via un apprentissage par manipulation passive. Cependant, notre robot n'a pas réalisé d'imitation en côte à côte. Cette situation est en fait la plus difficile des trois. Le partenaire agit dans son espace de travail tandis que le robot doit observer les actions et les transposer à son propre espace de travail. Une hypothèse de travail pour obtenir cette imitation en côte à côte est d'utiliser l'ambiguïté de la perception en jouant sur une mémoire attentionnelle et une confusion entre l'objet manipulé par l'humain et l'objet devant le robot. Du fait de l'ambiguïté de la perception et de la confusion des référentiels, le robot transposerait à son propre espace de travail les gestes du professeur humain suivant un référentiel centré objet. Nous sommes en revanche allé plus loin du point de vue de l'interaction sociale. En effet, le modèle implémenté a permis de modifier le comportement du robot grâce à un renforcement positif ou négatif pouvant venir d'une expression faciale.
Les comportements d'interaction mis en place par notre architecture visent à reproduire la manière dont on peut interagir avec un enfant à un stade préverbal pour réaliser une interface Homme-robot plus "naturelle" pour un humain voulant modifier le comportement du robot. Cette interface entre l'Homme et le robot se situe à trois niveaux : physique avec l'apprentissage

passif, visuel lors des imitations en face à face, et social grâce à l'exploitation de signaux de renforcements pouvant être extraits d'une reconnaissance d'expressions faciales.

Le principal enjeu pour le futur sera maintenant de pouvoir engager et mélanger ces différents types d'interaction dans une même tâche. Nous relâcherions ainsi les contraintes sur l'humain pour permettre des interactions plus naturelles et intuitives. Il est important de noter qu'il ne s'agit pas de rechercher un effet "esthétique" mais de soulever deux aspects fondamentaux : la qualité de l'interaction et l'unification des "modèles de fonctionnement" au sein d'une même architecture. Afin d'améliorer la qualité de l'interaction, l'interface Homme-machine réalisée devrait laisser l'humain libre d'aborder une phase de travail avec le robot sous l'angle souhaité (démontrer, manipuler le robot en passif, vérifier l'action via l'imitation par le robot). Cette qualité d'interface "intuitive" permettrait de faire durer dans le temps les interactions et d'aller vers des apprentissages de tâches réellement complexes. Bien que notre modèle définisse une architecture bio-inspirée commune pour la construction des différents comportements d'interaction, ceux-ci ont essentiellement été testés de manière indépendante. Les travaux futurs devront porter sur l'évaluation et la complétion de notre architecture pour aboutir à une architecture pour robot autonome unifiant complètement ces différents "modèles de fonctionnement".

Durant cette thèse, nous avons aussi beaucoup discuté, grâce au modèle DM-PerAc, de l'intérêt de pouvoir commander notre robot directement en force (notre modèle considérant ainsi à la fois la position et la force comme étant les grandeurs de contrôle pertinentes). L'usage de bras électriques à commande en position et en vitesse nous a obligés à simuler la commande en force par le biais d'une intégration numérique coûteuse et incapable de tenir compte des vrais efforts qui s'exerçaient sur le bras. La physique du bras, que ce soit pour son contrôle bas niveau ou sa morphologie, a un impact très important sur la capacité de l'architecture de contrôle à apprendre en interaction avec un humain. Les bras Katana ont par exemple posé de nombreux problèmes de vision liés à leur morphologie avec une tourelle et deux coudes perpétuellement dans le champ de vision de la caméra et cachant souvent les objets à manipuler. Disposer de bras anthropomorphiques fixés en position épaule représente beaucoup plus qu'un simple intérêt de ressemblance avec l'humain. Cela fournirait à notre architecture un moyen de rendre les apprentissages et les reproductions beaucoup plus performantes sans nécessiter d'améliorer le modèle. Pour finir, l'absence de compliance active des bras Katana nous avait obligés à développer un système mesurant l'effort pour rendre le bras complètement passif en situation d'interaction physique. Malheureusement, cette solution rend très difficile le raffinement de la trajectoire : l'expérimentateur ayant alors du mal à démontrer plusieurs fois la même trajectoire. L'arrivée de torses humanoïdes TINO au laboratoire ETIS devrait résoudre un grand nombre de ces problèmes en couplant sur des bras anthropomorphes un système de contrôle hydraulique [Alfayad et al., 2011] permettant un contrôle en force avec des retours d'efforts sur chaque segment du bras. La mise à disposition de bras dotés de capacités de compliance active et passive apparaît comme une nécessité pour aller plus loin. Les prochains travaux devraient confirmer le lien fort qui existe entre capacités d'interaction de "haut niveau" et propriétés mécaniques en insistant sur l'importance de ne pas négliger l'intelligence du corps [Pitti and Pfeifer, 2012].

Bibliographie

Alami, R., Clodic, A., Montreuil, V., Sisbot, E. A., and Chatila, R. (2005). Task planning for human-robot interaction. sOc-EUSAI '05, pages 81–85, Grenoble, France. ACM. 52

Albouy, G., Sterpenich, V., Balteau, E., Vandewalle, G., Desseilles, M., Dang-Vu, T., Darsaud, A., Ruby, P., Luppi, P.-H., Degueldre, C., et al. (2008). Both the hippocampus and striatum are involved in consolidation of motor sequence memory. *Neuron*, 58(2) :261–272. 33

Albu-Schäffer, A., Ott, C., and Hirzinger, G. (2007). A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. *The International Journal of Robotics Research*, 26(1) :23–39. 24

Albus, J. (1975). A new approach to manipulator control : The cerebellar model articulation controller (CMAC). *Journal of Dynamic Systems, Measurement, and Control*, (SEPTEMBER) :220–227. 32

Albus, J. S., Branch, D. T., Donald, C., and Perkel, H. (1971). A theory of cerebellar function. *Mathematical Biosciences*, 10(1-2) :25–61. 31

Alexander, G. E., DeLong, M. R., and Strick, P. L. (1986). Parallel organization of functionally segregated circuits linking basal ganglia and cortex. *Annual Review of Neuroscience*, 9(1) :357–381. 32

Alfayad, S., Ouezdou, F. B., Namoun, F., and Gheng, G. (2011). High performance integrated electro-hydraulic actuator for robotics - Part I : Principle, prototype design and first experiments. *Sensors and Actuators A : Physical*, 169(1) :115–123. 162

Amari, S.-I. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27(2) :77–87. 16, 35, 82

Anderson, M. L. (2010). Neural reuse : A fundamental organizational principle of the brain. *Behavioral and Brain Sciences*, 33(4) :245–266. 34

Andry, P. (2002). *Apprentissage par imitation : vers la compréhension des intentions de l'autre ?* PhD thesis, Université de Cergy-Pontoise. 37

Andry, P., Gaussier, P., Moga, S., Banquet, J. P., and Nadel, J. (2001). The dynamics of imitation processes : from temporal sequence learning to implicit reward communication. *IEEE Trans. on Man, Systems and Cybernetics Part A : Systems and humans*, 31(5) :431–442. 55

Andry, P., Gaussier, P., Nadel, J., and Hirsbrunner, B. (2004). Learning Invariant Sensorimotor Behaviors : A Developmental Approach to Imitation Mechanisms. *Adaptive Behavior*, 12(2) :117–140. 16, 36, 37, 57, 58, 82, 88, 89, 92

Ansuini, C., Santello, M., Massaccesi, S., and Castiello, U. (2006). Effects of end-goal on hand shaping. *Journal of Neurophysiology*, 95(4) :2456–2465. 28

Aosaki, T., Kimura, M., and Graybiel, A. (1995). Temporal and spatial characteristics of tonically active neurons of the primate's striatum. *Journal of neurophysiology*, 73(3) :1234–1252. 32

Arbib, M. A. (1981). Perceptual structures and distributed motor control. In Brooks, V., editor, *Handbook of Physiology - The Nervous System II. Motor Control*, pages 1449–1480. American Physiological Society. 53

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robot. Auton. Syst.*, 57(5) :469–483. 26

Arkin, R. C. (1998). *Behavior-based robotics [electronic resource]*. MIT press. 128

Atkeson, C. G., Andreww, and Stefan Schaal, Z. (1997). Locally weighted learning. In *Artificial Intelligence Review*, pages 11–73. 25

Baddeley, A. D. and Hitch, G. (1983). Working memory. 33

Banquet, J. P., Gaussier, P., Quoy, M., Revel, A., and Burnod, Y. (2005). A hierarchy of associations in hippocampo-cortical systems : cognitive maps and navigation strategies. *Neural computation*, 17(6) :1339–84. 53

Banquet, J.-P. P., Gaussier, P., Dreher, J. C., Joulain, C., Revel, A., Gunther, W., C.Joulain, A.Revel, and W.Gunther (1997). Space-time, order and hierarchy in fronto-hippocamal system : A neural basis of personality. In Mathews, G., editor, *Cognitive Science Perspectives on Personality and Emotion*, chapter 4, pages 123–189. Elsevier Science BV. 18, 33, 53, 92

Baranes, A. and Oudeyer, P.-Y. P.-Y. (2010). Intrinsically Motivated Goal Exploration for Active Motor Learning in Robots : a Case Study. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pages 1766–1773, Taipei Taiwan, Province Of China. INRIA, France, IEEE. 124, 159

Becchio, C., Sartori, L., Bulgheroni, M., and Castiello, U. (2008). The case of Dr. Jekyll and Mr. Hyde : A kinematic study on social intention. *Consciousness and Cognition*, 17(3) :557–564. 29

Bechara, A., Damasio, H., and Damasio, A. R. (2000). Emotion, decision making and the orbitofrontal cortex. *Cerebral cortex (New York, N.Y. : 1991)*, 10(3) :295–307. 33

Beiser, D., Hua, S., and Houk, J. (1997). Network models of the basal ganglia. *Current opinion in neurobiology*, 7(2) :185–190. 32

Bellman, R. E. (1957). Dynamic programming. page 342. 22, 23, 124

Berger, T. W. and Thompson, R. F. (1978). Neuronal plasticity in the limbic system during classical conditioning of the rabbit nictitating membrane response. I. The hippocampus. *Brain research*, 145(2) :323–346. 33

Berret, B., Darlot, C., Jean, F., Pozzo, T., Papaxanthis, C., and Gauthier, J. P. (2008). The Inactivation Principle : Mathematical Solutions Minimizing the Absolute Work and Biological Implications for the Planning of Arm Movements. *PLoS Comput Biol*, 4(10) :e1000194. 23

Bhushan, N. and Shadmehr, R. (1999). Computational nature of human adaptive control during learning of reaching movements in force fields. *Biological Cybernetics*, 81(1) :39–60. 30, 34, 157

Bicho, E., Louro, L., and Erlhagen, W. (2010). Integrating verbal and nonverbal communication in a dynamic neural field architecture for human-robot interaction. *Frontiers in neurorobotics*, 4. 52

Bizzi, E., Hogan, N., Ivaldi, F. A. M., and Giszter, S. (1992). Does the nervous system use equilibrium-point control to guide single and multiple joint movements ? *Behavioral and Brain Sciences*, 15(Special Issue 04) :603–613. 155

Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., and Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological review*, 108(3) :624–652. 33

Boucenna, S. (2011). *De la reconnaissance des expressions faciales a une perception visuelle partagée : une architecture sensori-motrice pour amorcer une référenciation sociale d'objets, de lieux ou de comportements.* PhD thesis, University of Cergy-Pontoise. 96

Boucenna, S., Gaussier, P., Andry, P., and Hafemeister, L. (2010a). Imitation as a Communication Tool for Online Facial Expression Learning and Recognition. In *IEEE/RSJ International Conference on Robots and Systems (IROS)*, pages 5323–5328. 96

Boucenna, S., Gaussier, P., Hafemeister, L., and Bard, K. (2010b). Autonomous Development of Social Referencing Skills. In *From Animals to Animats 11*, volume 6226, pages 628–638. Springer-Verlag Berlin, Heidelberg ©2010. 40, 91

Boucher, J.-D. and Dominey, P. F. (2006). Perceptual-motor sequence learning via human-robot interaction. *SAB 2006 LNAI 4095*, pages 224–235. 51, 52

Brooks, R. (1986). A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1) :14–23. 128

Bryson, J. J. (2000). Hierarchy and Sequence vs. Full Parallelism in Action Selection. In Meyer, J. A., Berthoz, A., Floreano, D., Roitblat, H., and Wilson, S., editors, *From {A}nimals to {A}nimats 6 : {P}roceedings of the {S}ixth {I}nternational {C}onference on {S}imulation of {A}daptive {B}ehavior*, pages 147–156, Cambridge, MA. MIT Press. 128

Bryson, J. J. (2012). Structuring intelligence : the role of hierarchy, modularity and learning in generating intelligent behaviour. In McFarland, D., Stenning, K., and McGonigle-Chalmers, M., editors, *The complex mind*, pages 126–143. Palgrave Macmillan, Basingstoke. 128

Bullock, D., Fiala, J. C., and Grossberg, S. (1994). A neural model of timed response learning in the cerebellum. *Neural Networks*, 7(6–7) :1101–1114. 33

BULLOCK, D. and GROSSBERG, S. (1989). Chapter 11 Vite and Flete : Neural Modules for Trajectory Formation and Postural Control. In Hershberger, W., editor, *Volitional Action*, volume 62 of *Advances in Psychology*, book chapter/section 11, pages 253–297. Elsevier. 37

Bullock, D., Grossberg, S., and Guenther, F. H. (1993). A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm. *J. Cognitive Neuroscience*, 5 :408–435. 39

Bush, G., Luu, P., and Posner, M. (2000). Cognitive and emotional influences in anterior cingulate cortex. *Trends in cognitive sciences*, 4(6) :215–222. 33

Byrne, R. W. and Russon, A. E. (1998). Learning by imitation : a hierarchical approach. *Behavioral and Brain Sciences*, 21(5) :667–84 ; discussion 684–721. 55

Calinon, S. and Billard, A. (2007). Incremental Learning of Gestures by Imitation in a Humanoid Robot. In *Proceedings of the {ACM/IEEE} International Conference on Human-Robot Interaction ({HRI})*, pages 255–262. 28

Calinon, S., D'halluin, F., Caldwell, D. G., and Billard, A. (2009). Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. In *Proceedings of 2009 IEEE International Conference on Humanoid Robots*, pages 582–588. 28, 41, 50, 58, 141, 144, 151

Calinon, S., D'halluin, F., Sauser, E., Caldwell, D., and Billard, A. (2010a). Learning and reproduction of gestures by imitation : An approach based on Hidden Markov Model and Gaussian Mixture Regression. *IEEE Robotics and Automation Magazine*, 17(2) :44–54. 26

Calinon, S., Guenter, F., and Billard, A. (2007). On Learning, Representing and Generalizing a Task in a Humanoid Robot. *IEEE transactions on systems, man and cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, 37(2) :286–298. 16, 19, 26, 58

Calinon, S., Sardellitti, I., and Caldwell, D. G. (2010b). Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In *Proc. {IEEE/RSJ} Intl Conf. on Intelligent Robots and Systems ({IROS})*, pages 249–254, Taipei, Taiwan. 26

Carpenter, G., Grossberg, S., Markuzon, N., Reynolds, J., and Rosen, D. (1992). Fuzzy ARTMAP : A neural network architecture for incremental supervised learning of analog multidimensional maps. *Neural Networks,*, 3(5) :698–713. 39

Carpenter, G. A. and Grossberg, S. (2002). Adaptive resonance theory (ART). In *The handbook of brain theory and neural networks*, pages 79–82. MIT Press, Cambridge, MA, USA. 42

Carrillo, R. R., Ros, E., Boucheny, C., and Coenen, O. J. (2008). A real-time spiking cerebellum model for learning robot control. *BioSystems*, 94(1) :18–27. 32

Cederborg, T., Li, M. L. M., Baranes, A., and Oudeyer, P.-Y. (2010). Incremental local online Gaussian Mixture Regression for imitation learning of multiple tasks. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. 28

Chaminade, T., Oztop, E., Cheng, G., and Kawato, M. (2008). From self-observation to imitation : visuomotor association on a robotic hand. *Brain research bulletin*, 75(6) :775–784. 58

Chiaverini, S., Siciliano, B., and Villani, L. (1999). A survey of robot interaction control schemes with experimental comparison. 24

Chiovetto, E., Berret, B., and Pozzo, T. (2010). Tri-dimensional and triphasic muscle organization of whole-body pointing movements. *Neuroscience*, 170(4) :1223–1238. 155

Clark, G., McCORMICK, D., Lavond, D., and Thompson, R. (1984). Effects of lesions of cerebellar nuclei on conditioned behavioral and hippocampal neuronal responses. *Brain research*, 291(1) :125–136. 31

Clodic, A., Cao, H., Alili, S., Montreuil, V., Alami, R., and Chatila, R. (2009). SHARY : A Supervision System Adapted to Human-Robot Interaction. In Khatib, O., Kumar, V., and Pappas, G., editors, *Experimental Robotics*, volume 54 of *Springer Tracts in Advanced Robotics*, pages 229–238. Springer Berlin Heidelberg. 52

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *J. Artif. Int. Res.*, 4(1) :129–145. 27

Cuperlier, N., Quoy, M., and Gaussier, P. (2007). Neurobiologically inspired mobile robot navigation and planning. *Front Neurorobotics*, 1 :3. 53

Cuperlier, N., Quoy, M., Giovannangeli, C., Gaussier, P., and Laroque, P. (2006). Transition Cells for Navigation and Planning in an Unknown Environment. volume 4095 of *Lecture Notes in Computer Science*, book part (with own title) 24, pages 286–297. Springer Berlin Heidelberg. 18

Daucé, E., Quoy, M., and Doyon, B. (2002). Resonant spatiotemporal learning in large random recurrent networks. *Biological Cybernetics*, 87(3) :185–198. 51

De Groot, A. D. (1978). *Thought and Choice in Chess*. Mouton De Gruyter, 2nd edition (June 1978). 150, 159

de Rengervé, A., Andry, P., and Gaussier, P. (2011). A Neural Network Generating Force Command for Motor Control of a Robotic Arm. In *International Workshop on bio-inspired robots*, Nantes, France.

de Rengervé, A., Andry, P., and Gaussier, P. (2013a). Autonomous Development of Gesture and Action Imitation as a Result of Sensorimotor Learning and Planning in a Brain Based Architecture.

de Rengervé, A., Andry, P., and Gaussier, P. (2013b). On-line Learning and Control of different kinds of attraction basins for development of sensory-motor control strategies. *Biological Cybernetics*, page soumis.

de Rengervé, A., Boucenna, S., Andry, P., and Gaussier, P. (2010a). Emergent Imitative Behavior on a Robotic Arm Based on Visuo-Motor Associative Memories. In *IEEE/RSJ International Conference on Intelligent Robots and systems (IROS'10)*, pages 1754–1759, Taipei, Taiwan. 41, 92

de Rengerve, A., Braud, R., Andry, P., and Gaussier, P. (2012). Behavior Adaptation from Negative Social Signal Based on Goal Awareness. In *2012 IEEE International Conference on Development and Learning (ICDL) - Epigenetics and Robotics (Epirob)*, pages 1–6, San Diego, CA, USA.

de Rengervé, A., D'halluin, F., Andry, P., Gaussier, P., and Billard, A. (2010b). A study of two complementary encoding strategies based on learning by demonstration for autonomous navigation task. In *Proceedings of the tenth international conference on Epigenetic Robotics*, pages 105–112, Orenas Slott, Sweden.

de Rengervé, A., Hanoune, S., Andry, P., Quoy, M., and Gaussier, P. (2013c). Building Specific Contexts for On-Line Learning of Dynamical Tasks through Non-verbal Interaction. In *2013 IEEE International Conference on Development and Learning (ICDL) - Epigenetics and Robotics (Epirob)*, pages 1–6, Osaka, Japan.

de Rengerve, A., Hirel, J., Andry, P., Quoy, M., and Gaussier, P. (2011). On-line learning and planning in a pick-and-place task demonstrated through body manipulation. In *IEEE International Conference on Development and Learning (ICDL) and on Epigenetic Robotics (Epirob), 2011*, volume 2, pages 1–6, Frankfurt am Main, Germany. IEEE.

de Rengervé, A., Hirel, J., Quoy, M., Andry, P., and Gaussier, P. (2011). A simple neural network controller merging different behaviors for collector robots. In *International Workshop on bio-inspired robots*, Nantes, France.

Demiris, Y. (2002). Imitation, Mirror Neurons, and the Learning of Movement Sequences. In *the International Conference on Neural Information Processing (ICONIP-2002)*, pages 111–115, Singapore. 57

Demiris, Y. and Johnson, M. (2003). Distributed, predictive perception of actions : a biologically inspired robotics architecture for imitation and learning. *Connect. Sci.*, 15(4) :231–243. 58

Detorakis, G. I. and Rougier, N. P. (2012). A Neural Field Model of the Somatosensory Cortex : Formation, Maintenance and Reorganization of Ordered Topographic Maps. *PLoS ONE*, 7(7) :e40257. 58

Dollé, L., Sheynikhovich, D., Girard, B., Chavarriaga, R., and Guillot, A. (2010). Path planning versus cue responding : a bio-inspired model of switching between navigation strategies. *Biological Cybernetics*, 103(4) :299–317. 32, 124, 158, 159

Dominey, P. and Boucher, J. (2005). Learning to talk about events from narrated video in a construction grammar framework. *Artificial Intelligence*, 167(1-2) :31–61. 51

Dominey, P. F. (1995). Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biological Cybernetics*, 73(3) :265–274. 51

Dominey, P. F., Mallet, A., and Yoshida, E. (2007). Real-time cooperative behavior acquisition by a humanoid apprentice. 52

Doya, K. (1999). What are the computations of the cerebellum, the basal ganglia and the cerebral cortex ? *Neural Networks*, 12(7-8) :961–974. 32, 34

Drescher, G. L. (1987). A mechanism for early Piagetian learning. In *AAAI'87*, pages 290–294. AAAI Press. 160

Drescher, G. L. (1991). *Made-up minds : a constructivist approach to artificial intelligence*. MIT Press, Cambridge, MA, USA. 160

Droniou, A., Ivaldi, S., Padois, V., and Sigaud, O. (2012). Autonomous Online Learning of Velocity Kinematics on the iCub : a Comparative Study. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, page To appear, Vilamoura, Portugal. 26

D'halluin, F., de Rengervé, A., Lagarde, M., Gaussier, P., Billard, A., and Andry, P. (2010). A state-action neural network supervising navigation and manipulation behaviors for complex task reproduction. In *Proceedings of the tenth international conference on Epigenetic Robotics*, pages 165–166, Orenas Slott, Sweden.

Eccles, J. C. et al. (1967). The cerebellum as a neuronal machine. *Electroencephalography and Clinical Neurophysiology*, 26(4) :451–452. 31

Eichenbaum, H., Otto, T., and Cohen, N. J. (1994). Two functional components of the hippocampal memory system. *Behavioral and Brain Sciences*, 17(03) :449–472. 32

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., Tang, C., and Rasmussen, D. (2012). A large-scale model of the functioning brain. *Science*, 338(6111) :1202–1205. 34

Feinman, S. (1982). Social referencing in infancy. *MerrillPalmer Quarterly*, 28(4) :445–470. 91, 96

Feldman, A. G. (1966). Functional tuning of the nervous system with control of movement or maintenance of a steady posture. II. Controllable parameters of the muscle. *Biophysics*, 11(3) :565–578. 155

Feldman, A. G., Adamovich, S. V., Ostry, D. J., and Flanagan, J. R. (1990). The Origin of Electromyograms - Explanations Based on the Equilibrium Point Hypothesis. In Winters, J. and Woo, S.-Y., editors, *Multiple Muscle Systems*, pages 195–213. Springer New York. 155

Feldman, A. G. and Levin, M. F. (2009). The Equilibrium-Point Hypothesis – Past, Present and Future Progress in Motor Control. In Sternad, D., editor, *Progress in Motor Control*, volume 629 of *Advances in Experimental Medicine and Biology*, book part (with own title) 38, pages 699–726. Springer US. 155

Fix, J., Rougier, N., and Alexandre, F. (2010). A Dynamic Neural Field Approach to the Covert and Overt Deployment of Spatial Attention. *Cognitive Computation*, 3(1) :279–293. 36, 88

Flash, T. (1987). The control of hand equilibrium trajectories in multi-joint arm movements. *Biological Cybernetics*, 57(4) :257–274. 24

Flash, T. and Hogan, N. (1985). The coordination of arm movements : an experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7) :1688–1703. 23

Fourneret, P. and Jeannerod, M. (1998). Limited conscious monitoring of motor performance in normal subjects. Technical Report 11, Institut des Sciences Cognitives, UPR 9075-CNRS, Lyon. fourneret@lyon151.inserm.fr. 92

Franz, M. O., Schölkopf, B., Mallot, H. A., and Bülthoff, H. H. (1998). Learning view graphs for robot navigation. *Autonomous Robots*, 5(1) :111–125. 52

Fukuyori, I., Nakamura, Y., Matsumoto, Y., and Ishiguro, H. (2008). Flexible Control Mechanism for Multi-DOF Robotic Arm Based on Biological Fluctuation. *From Animals to Animats 10*, pages 22–31. 16, 40, 62, 87, 149, 155

Fuster, J. (2008). *The Prefrontal Cortex*. Elsevier Science. 33

Gallese, V. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12) :493–501. 111

Gallese, V., Fadiga, L., Fogassi, L., and Rizzolatti, G. (1996). Action recognition in the premotor cortex. *Brain*, 119 :593–609. 56, 57, 111, 152

Ganesh, G. and Albu-Sch, A. (2010). Biomimetic motor behavior for simultaneous adaptation of force , impedance and trajectory in interaction tasks. In *Robotics and Automation ICRA 2010 IEEE International Conference on (2010)*, pages 2705–2711. IEEE. 87

Gaussier, P., Moga, S., Banquet, J. P., and Quoy, M. (1998). From Perception-Action loops to imitation processes : A bottom-up approach of learning by imitation. *Applied Artificial Intelligence*, 1(7) :701–727. 33, 36, 37, 57, 92

Gaussier, P., Revel, A., Banquet, J. P., and Babeau, V. (2001). From view cells and place cells to cognitive map learning : processing stages of the hippocampal system. *Biological Cybernetics*. 53

Gaussier, P., Revel, A., Banquet, J. P., and Babeau, V. (2002). From view cells and place cells to cognitive map learning : processing stages of the hippocampal system. *Biological Cybernetics*, 86(1) :15–28. 18, 93

Gaussier, P. and Zrehen, S. (1995). PerAc : A Neural Architecture to Control Artificial Animals. *Robotics and Autonomous Systems*, 16(2-4) :291–320. 17, 40, 44, 151

Gauthier, J.-P., Berret, B., and Jean, F. (2010). A biomechanical inactivation principle. *Proceedings of the Steklov Institute of Mathematics*, 268(1) :93–116. 23

Georgiou, I., Becchio, C., Glover, S., and Castiello, U. (2007). Different action patterns for cooperative and competitive behaviour. *Cognition*, 102(3) :415–433. 29

Georgopoulos, A., Schwartz, A., and Kettner, R. (1986). Neuronal population coding of movement direction. *Science*, 233(4771) :1416–1419. 31, 35, 37

Gilbert, P. F. C. (1974). A theory of memory that explains the function and structure of the cerebellum. *Brain Research*, 70(1) :1–18. 31

Giovannangeli, C. (2007). *Navigation autonome bio-inspirée en environnement intérieur et extérieur : Apprentissages sensori-moteurs et planification dans un cadre interactif*. PhD thesis, Université de Cergy-Pontoise. 50

Giovannangeli, C. and Gaussier, P. (2010). Interactive Teaching for Vision-Based Mobile Robots : A Sensory-Motor Approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 40(1) :13–28. 50, 131, 142

Giovannangeli, C., Gaussier, P., and Désilles, G. (2006). Robust Mapless Outdoor Vision-based Navigation. In *IEEE/RSJ International Conference on Intelligent Robots and systems*, Beijing, China. IEEE. 17, 47

Goldman-Rakic (1987). Circuitry of primate prefrontal cortex and regulation of behaviour by representational memory. In F, P. and Mountcastle, V., editors, *Handbook of Physiology : The Nervous System*, pages 373–417. American Physiological Society. 33

Gottlieb, G. L. (2000). A test of torque-control and equilibrium-point models of motor control. *Human Movement Science*, 19(6) :925–931. 155

Gottlieb, G. L. (2001). Influence of Strategy on Muscle Activity During Impact Movements. *Journal of Motor Behavior*, 33(3) :235–242. 155

Greve, D., Grossberg, S., Guenther, F., and Bullock, D. (1993). Neural representations for sensory-motor control, I : Head-centered 3-D target positions from opponent eye commands. *Acta Psychologica*, 82(1-3) :115–138. 39

Grossberg, S., Guenther, F. H., Bullock, D., and Greve, D. N. (1993). Neural representations for sensory-motor control, II : Learning a head-centered visuomotor representation of 3-D target position. *Neural Networks*, 6(1) :43–67. 39

Guenther, F. H., for Adaptive Systems, B. U. C., of Cognitive, B. U. D., and Systems, N. (1993). *Neural Representations for Sensory-motor Control, III : Learning a Body-centered Representation of 3-D Target Position*. Technical report CAS/CNS. Boston University, Center for Adaptive Systems and Department of Cognitive and Neural Systems. 39

Guitton, J. (2010). *Architecture hybride pour la planification d'actions et de d{é}placements*. These, Universit{é} Paul Sabatier - Toulouse III ; ISAE. 52

Gurney, K., Prescott, T. J., and Redgrave, P. (2001). A computational model of action selection in the basal ganglia. I. A new functional anatomy. *Biological Cybernetics*, 84(6) :401–410. 32

Hanoune, S., Quoy, M., and Gaussier, P. (2012). An architecture for online chunk learning and planning in complex navigation and manipulation tasks. In *IEEE International Conference on Development and Learning (ICDL) and on Epigenetic Robotics (Epirob), 2012*, page Submitted. 150, 159

Harnad, S. (1990). The symbol grounding problem. *Physica D : Nonlinear Phenomena*, 42(1-3) :335 – 346. 13

Harris, C. M. and Wolpert, D. M. (1998). Signal-Dependent Noise Determines Motor Planning. *Nature*, 394 :780. 23

Hasselmo, M. E. (2005). A model of prefrontal cortical mechanisms for goal-directed behavior. *Journal of Cognitive Neuroscience*, 17(7) :1115–1129. 33, 53

Hasselmo, M. E. and Schnell, E. (1994). Laminar selectivity of the cholinergic suppression of synaptic transmission in rat hippocampal region CA1 : computational modeling and brain slice physiology. *The Journal of neuroscience*, 14(6) :3898–3914. 33

Hasson, C., Boucenna, S., Gaussier, P., and Hafemeister, L. (2010). Using Emotional Interactions for Visual Navigation Task Learning. In *Proceedings of the International Conference on Kansei Engineering and Emotion Research*, pages 1578–1587, {P}aris {F}rance. 91

Hatze, H. and Buys, J. D. (1977). Energy-optimal controls in the mammalian neuromuscular system. *Biological Cybernetics*, 27(1) :9–20. 23

Hersch, M. and Billard, A. (2006). A Biologically-Inspired Model of Reaching Movements. In *In Proceedings of the 2006 IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, Pisa*. 38

Heyes, C. (2010). Where do mirror neurons come from ? *Neuroscience & Biobehavioral Reviews*, 34(4) :575–583. 57

Hirel, J. (2011). *Codage hippocampique par transitions spatio-temporelles pour l'apprentissage autonome de comportements dans des tâches de navigation sensori-motrice et de planification en robotique*. PhD thesis, Université de Cergy-Pontoise. 93, 94, 123

Hirel, J., Gaussier, P., and Quoy, M. (2010). Model of the Hippocampal Learning of Spatio-temporal Sequences. In *Artificial Neural Networks – ICANN 2010*, volume 6354, pages 345–351. 54, 93

Hirel, J., Gaussier, P., and Quoy, M. (2011). Biologically inspired neural networks for spatio-temporal planning in robotic navigation tasks. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 1627–1632. 53

Hirel, J., Gaussier, P., Quoy, M., Banquet, J.-P., Save, E., and Poucet, B. (2013). The hippocampo-cortical loop : Spatio-temporal learning and goal-oriented planning in navigation. *Neural Networks*, 43(0) :8–21. 54, 93

Hoffmann, H., Pastor, P., Park, D.-H., and Schaal, S. (2009). biologically-inspired dynamical systems for movement generation : automatic real-time goal adaptation and obstacle avoidance. In *international conference on robotics and automation (icra2009)*. 26

Hogan, N. (1984a). An organizing principle for a class of voluntary movements. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 4(11) :2745–2754. 23, 24, 151

Hogan, N. (1984b). Impedance Control : An Approach to Manipulation. In *American Control Conference, 1984*, pages 304–313. Department of Mechanical Engineering, Laboratory of Manufacturing and Productivity, Massachusetts Institute of Technology, Cambridge, MA 02139, IEEE. 17

Holmes, G. (1939). The cerebellum of man. *Brain*, 62(1) :1–30. 31

Iacoboni, M., Molnar-Szakacs, I., Gallese, V., Buccino, G., Mazziotta, J. C., and Rizzolatti, G. (2005). Grasping the intentions of others with one's own mirror neuron system. *Plos Biology*, 3(3) :529–535. 111

Ijspeert, A., Nakanishi, J., and Schaal, S. (2001). Trajectory formation for imitation with nonlinear dynamical systems. *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, 2 :752–757 vol.2. 26

Ijspeert, A. J., Nakanishi, J., and Schaal, S. (2003). Learning attractor landscapes for learning motor primitives. In *advances in neural information processing systems 15*, pages 1547–1554. cambridge, ma : mit press. 26, 58

Iossifidis, I. and Schoner, G. (2004). Autonomous reaching and obstacle avoidance with the anthropomorphic arm of a robotic assistant using the attractor dynamics approach. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 5, pages 4295–4300 Vol.5. Inst. fur Neuroinformatik, Ruhr-Univ., Bochum, Germany, IEEE. 16, 36, 82

Iossifidis, I. and Schoner, G. (2006). Dynamical Systems Approach for the Autonomous Avoidance of Obstacles and Joint-limits for an Redundant Robot Arm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'06)*, pages 580–585. Institut fur Neuroinformatik, Ruhr-Universitat Bochum, IEEE. 39

Ivaldi, S., Sigaud, O., Berret, B., and Nori, F. (2012). From humans to humanoids : The optimal control framework. *Paladyn*, 3(2) :75–91. 22

Jacob, P. and Jeannerod, M. (2005). The motor theory of social cognition : a critique. *Trends in cognitive sciences*, 9(1) :21–25. 28, 55, 111

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology. 51

Jauffret, A., Cuperlier, N., Gaussier, P., and Tarroux, P. (2013). From Self-Assessment to Frustration, A Small Step Towards Autonomy in Robotic Navigation. *Frontiers in Neurorobotics*, 7(16). 124, 158, 159, 161

Jeannerod, M. (1999). To act or not to act. Perspectives on the representation of actions. *Quaterly Journal of Experimental Psychology*, 52A :1–29. 92

Jiménez-Fabián, R. and Verlinden, O. (2011). Review of control algorithms for robotic ankle systems in lower-limb orthoses, prostheses, and exoskeletons. *Medical Engineering & Physics*, 34(4) :397–408. 24

Johnson, J. S., Spencer, J. P., Luck, S. J., and Schöner, G. (2009). A dynamic neural field model of visual working memory and change detection. *Psychological Science*, 20(5) :568–77. 36, 88

Jones, S. S. (2009). The development of imitation in infancy. *Philosophical Transactions of the Royal Society B : Biological Sciences*, 364(1528) :2325–2335. 57

Kawato, M., Furukawa, K., and Suzuki, R. (1987). A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57(3) :169–185. 31, 32

Khamassi, M., Quilodran, R., Enel, P., Procyk, E., and Dominey, F. (2010). A computational model of integration between reinforcement learning and task monitoring in the prefrontal cortex. *From Animals to Animats*, 6226(11) :424–434. 40

Klanke, S., Vijayakumar, S., and Schaal, S. (2007). A Library for Locally Weighted Projection Regression. *Journal of Machine Learning Research*, 9 :623–626. 44

Kugiumutzakis, G. (1998). Neonatal imitation in the intersubjective companion space. *Intersubjective communication and emotion in early ontogeny*, pages 63–88. 57

Kühn, R. and Hemmen, J. (1991). Temporal Association. In Domany, E., Hemmen, J., and Schulten, K., editors, *Models of Neural Networks*, Physics of Neural Networks, pages 213–280. Springer Berlin Heidelberg. 51

Lagarde, M. (2010). *Apprentissage de nouveaux comportement : vers le développement épigénétique d'un robot autonome*. PhD thesis, Université de Cergy-Pontoise. 51

Lagarde, M., Andry, P., and Gaussier, P. (2007). The Role of Internal Oscillators for the One-Shot Learning of Complex Temporal Sequences. In de Sa, J. M., Alexandre, L. A., Duch, W., and Mandic, D., editors, *Artificial Neural Networks – ICANN 2007*, volume 4668 of *LNCS*, pages 934–943. Springer. 54, 58

Lagarde, M., Andry, P., Gaussier, P., Boucenna, S., and Hafemeister, L. (2010). Proprioception and Imitation : On the Road to Agent Individuation. In Sigaud, O. and Peters, J., editors, *From Motor Learning to Interaction Learning in Robots*, volume 264, book part (with own title) 3, pages 43–63. Springer Berlin Heidelberg, Berlin, Heidelberg. 47, 93

Lagarde, M., Andry, P., Gaussier, P., and Giovannangeli, C. (2008). Learning New Behaviors : Toward a Control Architecture Merging Spatial and Temporal Modalities. In *Workshop on Interactive Robot Learning - International Conference on Robotics : Science and Systems (RSS 2008)*. 54

Lallee, S., Warneken, F., and Dominey, P. F. (2009). Learning to Collaborate by Observation. In *Proceedings of the 9th International Conference on Epigenetic Robotics*, pages 219–220. 52

Lallee, S., Yoshida, E., Mallet, A., Nori, F., Natale, L., Metta, G., Warneken, F., and Dominey, P. F. (2010). Human-Robot Cooperation Based on Interaction Learning. *From Motor Learning to Interaction Learning in Robots*, 264 :491–536. 52

Landmann, C. (2007). *Le cortex préfrontal et la dopamine striatale dans l'apprentissage guidé par la récompense*. PhD thesis, UNIVERSITE PARIS 6. 33

Lewkowicz, D., Delevoye-Turrell, Y., Bailly, D., Andry, P., and Gaussier, P. (2013). Reading Motor Intention through Mental Imagery. *Adaptive Behavior*, page accepted. 29

Li, Y., Kurata, S., Morita, S., Shimizu, S., Munetaka, D., and Nara, S. (2008). Application of chaotic dynamics in a recurrent neural network to control : hardware implementation into a novel autonomous roving robot. *Biological Cybernetics*, 99(3) :185–196. 51

Ljung, L. and Söderström, T. (1983). *Theory and practice of recursive identification*. MIT press Cambridge, MA. 26

Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics : a survey. *Connection Science*, 15(4) :151–190. 15

Maes, P. and Brooks, R. (1990). Learning to Coordinate Behaviors. In *AAAI Proceedings*, pages 796–802. 128, 160

Maillard, M., Gapenne, O., Hafemeister, L., and Gaussier, P. (2005). Perception as a Dynamical Sensori-Motor Attraction Basin. *Advances in Artificial Life*, 3630 :37–46. 43, 161

Marr, D. (1969). A theory of cerebellar cortex. *The journal of physiology*, 202(2) :437–470. 31

Martinet, L.-E., Sheynikhovich, D., Benchenane, K., and Arleo, A. (2011). Spatial Learning and Action Planning in a Prefrontal Cortical Network Model. *PLoS Comput Biol*, 7(5) :e1002045. 53

Maturana, H. and Varela, F. (1992). *The tree of knowledge : the biological roots of human understanding*. Shambhala. 16

McCormick, D. and Thompson, R. (1984). Cerebellum : essential involvement in the classically conditioned eyelid response. *Science*, 223(4633) :296–299. 31

McIntyre, J. and Bizzi, E. (1993). Servo Hypotheses for the Biological Control of Movement. *Journal of Motor Behavior*, 25(3) :193–202. 156

Meltzoff, A. N. and Moore, M. K. (1977). Imitation of facial and manual gestures by human neonates. *Science*, 198(4312) :75–78. 57

Metta, G., Sandini, G., Natale, L., Craighero, L., and Fadiga, L. (2006). Understanding mirror neurons : A bio-robotic approach. *Interaction Studies*, 7(2) :197–232. 57

Meyer, J.-A. (1996). Artificial Life and the Animat Approach to Artificial Intelligence. In Boden, M., editor, *Artificial Intelligence*, pages 325–354. Academic Press. 18, 54

Miller, E. K., Freedman, D. J., Wallis, J. D., Miller, E. K., Freedman, D. J., and Wallis, J. D. (2002). The prefrontal cortex : categories, concepts and cognition. *Philosophical Transactions of the Royal Society of London. Series B : Biological Sciences*, 357(1424) :1123–1136. 33

Miller, G. (1956). The Magical Number Seven, Plus or Minus Two : Some Limits on Our Capacity for Processing Information. *Psychological Review*, 63(2) :81–97. 150, 159

Miller, W., I. (1987). Sensor-based control of robotic manipulators using a general learning algorithm. *IEEE Journal on Robotics and Automation*, 3(2). 32

Miyamoto, H. and Kawato, M. (1998). A tennis serve and upswing learning robot based on bi-directional theory. *Neural Networks*, 11(7-8) :1331–1344. 24

Montesano, L., Lopes, M., Bernardino, A., and Santos-Victor, J. (2008). Learning Object Affordances : From Sensory Motor Coordination to Imitation. *IEEE Transactions on Robotics*, 24(1) :15–26. 58

Muller, R. U., Stead, M., and Pach, J. (1996). The hippocampus as a cognitive graph. *The Journal of general physiology*, 107(6) :663–94. 52

Nadel, J. (1986). *Imitation et communication entre jeunes enfants*. Presse Universitaire de France, Paris. 17

Nadel, J., Prepin, K., and Okanda, M. (2005). Experiencing contigency and agency : first step toward self-understanding ? *Interaction Studies*, 2 :447–462. 55

Nagai, Y., Nakatani, A., and Asada, M. (2010). How a robot's attention shapes the way people teach. In *Proceedings of the 10th International Conference on Epigenetic Robotics*, pages 81–88. 29

Nakano, E., Imamizu, H., Osu, R., Uno, Y., Gomi, H., Yoshioka, T., and Kawato, M. (1999). Quantitative examinations of internal representations for arm trajectory planning : minimum commanded torque change model. *Journal of neurophysiology*, 81(5) :2140–55. 23

Natale, L., Nori, F., Sandini, G., and Metta, G. (2007). Learning precise 3D reaching in a humanoid robot. In *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pages 324–329. Italian Inst. of Technol., Genova, IEEE. 26

Nehaniv, C. L. and Dautenhahn, K. (2002). The correspondence problem. pages 41–61. MIT Press. 56, 92

Nelson, W. L. (1983). Physical principles for economies of skilled movements. *Biological Cybernetics*, 46(2) :135–147. 23

Nguyen, S. M. and Oudeyer, P.-Y. (2012). Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner. *Paladyn Journal of Behavioural Robotics*, 3(3) :136–146. 124, 158

Nielsen, T. I. (1963). Volition : A new experimental approach. *Scandinavian Journal of Psychology*, 4 :225–230. 92

Nishii, J. and Taniai, Y. (2009). Evaluation of trajectory planning models for arm-reaching movements based on energy cost. *Neural Computation*, 21(9) :2634–2647. 23

Nishimoto, R., Namikawa, J., and Tani, J. (2008). Learning Multiple Goal-Directed Actions Through Self-Organization of a Dynamic Neural Network Model : A Humanoid Robot Experiment. *Adaptive Behavior*, 16(2-3) :166–181. 57

Nurzaman, S., Matsumoto, Y., Nakamura, Y., Koizumi, S., and Ishiguro, H. (2009). Yuragi-based adaptive searching behavior in mobile robot : From bacterial chemotaxis to Levy walk. pages 806–811. 40

O'Keefe, J. and Nadel, L. (1979). Précis of O'Keefe & Nadel's The hippocampus as a cognitive map. *Behavioral and Brain Sciences*, 2(04) :487–494. 32, 47, 52

O'Reilly, R. C. (1997). The LEABRA model of neural interactions and learning in the neocortex. *Dissertation Abstracts International : Section B : The Sciences and Engineering*, 57(11-B) :6792. 34

O'Reilly, R. C. (1998). Six principles for biologically based computational models of cortical cognition. *Trends in cognitive sciences*, 2(11) :455–462. 34

O'Reilly, R. C., Hazy, T. E., and Herd, S. A. (2013). The Leabra cognitive architecture : How to play 20 principles with nature and win ! In *Oxford Handbook of Cognitive Science*. Oxford Univ Press. 34

Oudeyer, P.-Y. and Kaplan, F. (2004). Intelligent Adaptive Curiosity : a source of Self-Development. In *Lund University Cognitive Studies*, volume 117, pages 127–130. 87

Oztop, E., Kawato, M., and Arbib, M. (2006). Mirror neurons and imitation : a computationally guided review. *Neural networks : the official journal of the International Neural Network Society*, 19(3) :254–271. 57

Piaget, J. (1936). *La naissance de l'intelligence chez l'enfant*. Delachaux & Niestlé. 15, 160

Piaget, J. (1945a). *La formation du symbole chez l'enfant*. Delachaux et Niestle Editions. 17

Piaget, J. (1945b). *La formation du symbole chez l'enfant. Imitation, jeu et rêve. Image et représentation*. Neuchâtel ; Paris : Delachaux et Niestlé. 55

Pitti, A. and Pfeifer, R. (2012). *La révolution de l'intelligence du corps*. Manuella Editions. 162

Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F. (1962). *The mathematical theory of optimal processes*. Wiley, New York, NY. 22

Quoy, M., Banquet, J.-P., and Daucé, E. (2001). Learning and Control with Chaos : From Biology to Robotics. *Behavioral and Brain Sciences*, 24(5) :824–825. 51

Redgrave, P. and Gurney, K. (2006). The short-latency dopamine signal : a role in discovering novel actions ? *Nature Reviews Neuroscience*, 7(12) :967–975.

Redgrave, P., Prescott, T. J., and Gurney, K. (1999). The basal ganglia : a vertebrate solution to the selection problem ? *Neuroscience*, 89(4) :1009–1023. 32

Redish, A. D. and Touretzky, D. S. (1998). The role of the hippocampus in solving the Morris water maze. *Neural Computation*, 10(1) :73–111. 52

Reimann, H., Iossifidis, I., and Schoner, G. (2011). Autonomous movement generation for manipulators with multiple simultaneous constraints using the attractor dynamics approach. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5470–5477. Inst. fur Neuroinformatik, Ruhr-Univ. Bochum, Bochum, Germany, IEEE. 36

Reiss, M. and Taylor, J. G. (1991). Storing temporal sequences. *Neural Networks*, 4(6) :773–787. 33

Revel, A., Gaussier, P., Lepretre, S., and Banquet, J. P. (1998). Planification versus sensory-motor conditioning : what are the issues ? In *Proceedings of the fifth international conference on simulation of adaptive behavior on From animals to animats 5*, pages 129–138, Cambridge, MA, USA. MIT Press. 53

Rizzolatti, G. and Craighero, L. (2004). THE MIRROR-NEURON SYSTEM. *Annual Review of Neuroscience*, 27(1) :169–192. 57, 111, 152

Rizzolatti, G., Fabbri-Destro, M., and Cattaneo, L. (2009). Mirror neurons and their clinical relevance. *Nature Clinical Practice Neurology*, 5(1) :24–34. 56

Rosenbaum, D. A. and Jorgensen, M. J. (1992). Planning macroscopic aspects of manual control. *Human Movement Science*, 11(1-2) :61–69. 28

Rosenblatt, F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychol Rev.*, 65(6) :386–408. 32

Rougier, N. P. (2009). Implicit and explicit representations. *Neural Networks*, 22(2) :155–160. 36

Ruffman, T. (2001). Understanding a-Not-B Errors as a Function of Object Representation and Deficits in Attention Rather Than Motor Memories. *Behavioral and Brain Sciences*, 24(1) :61. 36

Sanes, J. N. and Jennings, V. A. (1984). Centrally programmed patterns of muscle activity in voluntary motor behavior of humans. *Experimental Brain Research*, 54(1) :23–32. 155

Sartenaer, O. (2010). Définir l'émergence. *Revue des Questions Scientifiques*, 181(3) :371–404. 16

Schaal, S. (1997). Learning from demonstration. In *advances in neural information processing systems 9*, pages 1040–1046. mit press. 25

Schaal, S. (2003). Dynamic movement primitives - a framework for motor control in humans and humanoid robots. In *the international symposium on adaptive motion of animals and machines*. 26

Schaal, S. and Atkeson, C. G. (1998). Constructive Incremental Learning from Only Local Information. *Neural Comput.*, 10(8) :2047–2084. 25

Schaal, S., Ijspeert, A., and Billard, A. (2004). *Computational approaches to motor learning by imitation*, pages 199–218. Number 1431. oxford university press. 58

Schmajuk, N. A. and DiCarlo, J. J. (1992). Stimulus configuration, classical conditioning, and hippocampal function. *Psychological Review*, 99(2) :268{\textendash}305. 150, 159

Schmidhuber, J. (1991). Curious model-building control systems. In *Proceedings of the International Joint Conference on Neural Networks Singapore*, volume 2, pages 1458–1463. IEEE, IEEE press. 87

Schoner, G. (1995). Dynamics of behavior : Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16(2-4) :213–245. 16, 35, 36, 49, 82, 88, 89

Schultz, W. (2000). Reward Processing in Primate Orbitofrontal Cortex and Basal Ganglia. *Cerebral Cortex*, 10(3) :272–283. 32

Shepherd, G. (2003). *The synaptic organization of the brain*. Oxford University Press, USA. 31

Short, M. W. and Cauraugh, J. H. (1997). Planning macroscopic aspects of manual control : end-state comfort and point-of-change effects. *Acta Psychologica*, 96(1-2) :133–147. 28

Short, M. W. and Cauraugh, J. H. (1999). Precision hypothesis and the end-state comfort effect. *Acta Psychologica*, 100(3) :243–252. 28

Spence, K. W. (1937). Experimental studies of learning and higher mental processes in infra-human primates. *Psychological Bulletin*, 34 :806–850. 55

Srinivasa, N., Bhattacharyya, R., Sundareswara, R., Lee, C., and Grossberg, S. (2012). A bio-inspired kinematic controller for obstacle avoidance during reaching tasks with real robots. *Neural Networks*, 35 :54–69. 39, 154

Steels, L. and Baillie, J.-C. (2003). Shared grounding of event descriptions by autonomous robots. *Robotics and Autonomous Systems*, 43(2-3) :163–173. 13

Sugahara, A., Nakamura, Y., Fukuyori, I., Matsumoto, Y., and Ishiguro, H. (2010). Generating Circular Motion of a Human-Like Robotic Arm Using Attractor Selection Model. *Journal ofRobotics andMechatronics*, 22(3) :315–321. 40

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning : An Introduction*. MIT press. 25, 124, 159

Taniai, Y. and Nishii, J. (2008). Optimality of Reaching Movements Based on Energetic Cost under the Influence of Signal-Dependent Noise. *NEURAL INFORMATION PROCESSING Lecture Notes in Computer Science*, 4984 :1091–1099. 23

Thelen, E., Schöner, G., Scheier, C., and Smith, L. B. (2001). The dynamics of embodiment : A field theory of infant perseverative reaching. *Behavioral and Brain Sciences*, 24(01) :1–34. 35, 36

Thomaz, A., Berlin, M., and Breazeal, C. (2005). An embodied computational model of social referencing. In *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pages 591–598, Nashville, TN, USA. IEEE. 91

Thomaz, A. L. and Breazeal, C. (2007). Asymmetric Interpretations of Positive and Negative Human Feedback for a Social Learning Agent. 111

Thomaz, A. L. and Cakmak, M. (2009). Learning about objects with human teachers. HRI '09, pages 15–22, La Jolla, California, USA. ACM. 29

Thorndike, E. L. (1898). Animal Intelligence : An Experimental Study of the Associative Processes in Animals. *Psychological Monographs*, 2(10) :1125–1127. 55

Thrun, S. and Mitchell, T. M. (1995). Lifelong robot learning. *Robotics and Autonomous Systems*, 15(1–2) :25–46. 128, 160

Todorov, E. (2004). Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9) :907–15. 24, 157

Todorov, E. (2006). Optimal Control Theory. *Environment and Planning C Government and Policy*, 4(2) :1–28. 23

Todorov, E. and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nat Neurosci*, 5(11) :1226–1235. 25, 58

Todorov, E. and Jordan, M. I. (2003). A Minimal Intervention Principle for Coordinated Movement. In *Advances in Neural Information Processing Systems*, pages 27–34. 24, 29, 58, 151

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55(4) :189–208. 18, 52

Tomasello, M., Carpenter, M., Call, J., Behne, T., and Moll, H. (2005). Understanding and sharing intentions : the origins of cultural cognition. *Behavioral and Brain Sciences*, 28(5) :675–691 ; discussion 691–735. 55, 111, 123

Tomasello, M., Davis-Dasilva, M., Camak, L., and Bard, K. (1987). Observational learning of tool-use by young chimpanzees. *Human Evolution*, 2(2) :175–183. 55

Trullier, O., Wiener, S. I., Berthoz, A., and Meyer, J. A. (1997). Biologically based artificial navigation systems : review and prospects. *Progress in neurobiology*, 51(5) :483–544. 52

Tyrrell, T. (1993). The Use of Hierarchies for Action Selection. *Adaptive Behavior*, 1(4) :387–420. 128

Uno, Y., Kawato, M., and Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint arm movement. *Biological Cybernetics*, 61(2) :89–101. 23

Uzgiris, I. C. (1999). Imitation as activity : its dev elopmental aspects. In *Imitation in infancy*, pages 187–206. Cambridge University Press. 55

Vijayakumar, S., D'souza, A., and Schaal, S. (2005). Incremental Online Learning in High Dimensions. *Neural Comput.*, 17(12) :2602–2634. 16, 26, 41, 44, 58, 151

Voicu, H. and Schmajuk, N. (2000). Exploration, Navigation and Cognitive Mapping. *Adaptive Behavior*, 8(3-4) :207–223. 52

Wada, Y. and Kawato, M. (2004). A via-point time optimization algorithm for complex sequential trajectory formation. *Neural Networks*, 17(3) :353–364. 24

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4) :279–292. 25, 128, 129

Whiten, A. and Ham, R. (1992). On the nature and evolution of imitation in the animal kingdom : Reappraisal of a century of research. *Advances in the study of behavior*, 21. 55

Widrow, B. and Hoff, M. E. (1960). Adaptive Switching Circuits. In *1960 {IRE} {WESCON} Convention Record, Part 4*, pages 96–104, New York. IRE. 42, 49, 145

Wilson, S. W. (1991). The animat path to AI. In Meyer, J. A. and Wilson, S. W., editors, *From animals to animats 1*, pages 15–21. MIT Press Cambridge, MA, USA, Citeseer. 54

Wolpert, D. M., Miall, R. C., and Kawato, M. (1998). Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9) :338–347. 31, 32, 157

Zacharias, F. (2012). *Knowledge Representations for Planning Manipulation Tasks*. Cognitive Systems Monographs. Springer Berlin Heidelberg. 52

Zajac, F. (1989). Muscle and tendon : properties, models, scaling, and application to biomechanics and motor control. *Critical reviews in biomedical engineering*, 17(4) :359–411. 155

Zukowgoldring, P. and Arbib, M. (2007). Affordances, effectivities, and assisted imitation : Caregivers and the directing of attention. *Neurocomputing*, 70(13-15) :2181–2193. 55

ANNEXE A

Comparaison entre le modèle PerAc et le modèle à mixture de Gaussiennes pour le contrôle d'un robot navigateur

de Rengervé, A., D'halluin, F., Andry, P., Gaussier, P., and Billard, A. (2010b). A study of two complementary encoding strategies based on learning by demonstration for autonomous navigation task. In *Proceedings of the tenth international conference on Epigenetic Robotics*, pages 105–112, Orenas Slott, Sweden

A study of two complementary encoding strategies based on learning by demonstration for autonomous navigation task

Antoine de Rengervé* **Florent D'halluin**** **Pierre Andry***
Philippe Gaussier* **Aude Billard****
*ETIS, CNRS ENSEA University Cergy-Pontoise F-95000 Cergy-Pontoise
**LASA, EPFL, CH-1015 Lausanne, SWITZERLAND

Abstract

Learning by demonstration is a natural and interactive way of learning which can be used by non-experts to teach behaviors to robots. In this paper we study two learning by demonstration strategies which give different answers about how to encode information and when to learn. The first strategy is based on artificial Neural Networks and focuses on reactive on-line learning. The second one uses Gaussian Mixture Models built on statistical features extracted off-line from several training datasets. A simple navigation experiment is used to compare the developmental possibilities of each strategy. Finally, they appear to be complementary and we will highlight that each one can be related to a specific memory structure in brain.

1. Introduction

Human development is clearly influenced by interactions performed during the whole life. One of the natural way to teach someone how to do something is simply to demonstrate what is expected from him. Programming by demonstration [Billard et al., 2008] is a key approach for development in autonomous robots. It does not require any technical knowledge from the teacher as it tries to provide the robot with learning abilities similar to those present in human beings. This is particularly interesting when dealing with robotic systems that must perform a wide range of tasks. Approaches requesting to program every possible behavior happen to be a dead-end.

In this work, demonstrations enable the robot to build behaviors based on the learning of sensorimotor associations. The system can then infers what to do when presented with some given sensory information (proprioception, vision). However, there are different ways to encode such sensorimotor associations, and different ways to learn them, which will influence the developmental capacities of the robot. In this paper, we will focus on two strategies that give distinct answers to the question of encoding and learning. The Neurocyber team of ETIS lab developed a system based on artificial Neural Networks that allows a user to teach a mobile robot how to navigate robustly using visuo-motor association [Lagarde et al., 2010]. Meanwhile, the LASA lab uses a statistical approach based on Gaussian Mixture Models to learn the sensorimotor coupling. This can be employed for teaching gestures to robotic arms or humanoid robots [Calinon et al., 2009].

A simple U-shaped navigation task has been chosen in order to compare these two approaches. The robotic platform used in this work, a Robulab from Robosoft, can select a direction to go forward. When the robot goes away from the right path, its direction can be modified by using a joystick. In order to compute its position, the robot can use a monocular pan moving camera. With pan rotation, the camera provides the robot with visual panorama for self-localization. An odometer can be used for recording the trajectory in the Cartesian space.

The Neural Network (NN) model and the Gaussian Mixture Models (GMM) that can perform navigation task will be respectively developped in Section 2 and Section 3. The presented experiment illustrates similarities and differences between these two approaches. These two systems enable a robot to learn actions in the context of learning by demonstration in interaction with a human teacher. Both systems are based on state-action associations but they do not learn and encode information in the same way. Section 4 discusses the consequences of these different encodings for the capacities of the system while considering memory-cost, adaptation through long time learning, interaction and quality of the trajectory. In Section 5, we discuss the complementary aspects of these two approaches and how they could be related to different kinds of memories as human cognition is concerned. They show similarities with the Hippocampus and the Neocortex as described in [McClelland et al., 1995].

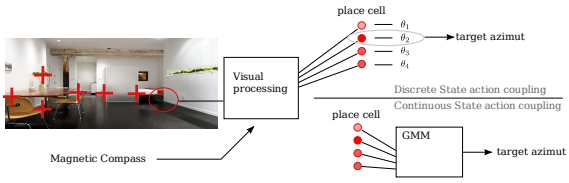


Figure 1: Overview of the system: Once the visual processing is done on data, it activates continuously the place cells neurons.

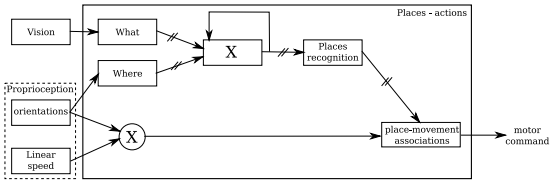


Figure 2: Model of place-movement associations learning. "what" and "where" information is extracted from vision and proprioception. They are merged and compressed in place codes allowing place recognition. These place codes are associated to proprioceptive informations.

2. Using Neural Networks for navigation

Neurocyber team has developed a controller for mobile robots which is able to associate visual information (a panorama of the environment) with self orientation (using a compass representing the direction of the actual movement). This controller is designed as a sensorimotor loop based on a neurobiological model testing some of the spatial properties of the hippocampus [Giovannangeli et al., 2006].

2.1 Place cells definitions

To be able to localize itself and navigate, the robot uses the recognition of place cells based on visual cues (Figure 1 and [Giovannangeli et al., 2006] for more details about visual features extraction). A place is defined by a constellation of visual features (landmark-azimuth couples) extracted from a panorama (Figure 3) compressed into a place code. The visual system extracts local views centered on points of interest (landmark recognition) that provides information of "what". A magnetic compass acting as proprioceptive information (spatial localization in the visual field) provides information of "where". The place code results of the merging of "what" and "where" information.

The merging of the information is performed in a product space (*i.e.* a matrix of product neurons $m_k(t)$ called *merging neurons*) defining a place code $M(t)$. More details about the definition of the place code and the merging neurons can be found

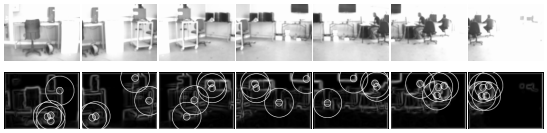


Figure 3: Example of visual features extraction on half visual panorama. The system computes a gradient on each view. 4 visual features are extracted from each gradient.

in [Lagarde et al., 2010]. The place-cell activities are built as the result from the computation of the distance between the learned place code and the current place code. The activity $PC_p(t)$ of the p^{th} place cell is:

$$PC_p(t) = \frac{1}{W_p}\left(\sum_{k=1}^{n_M} \omega_{kp}^{PC}(t) m_k(t)\right) \quad (1)$$

where $\omega_{kp}^{PC}(t)$ expresses the fact that the landmark-azimuth couple k (*i.e.* the k^{th} merging neuron which activity is $m_k(t)$) has been used to encode the place cell p. The number of couples used by the p^{th} place cell is given by $W_p = \sum_{k=1}^{n_M} \omega_{kp}^{PC}$, with n_M the number of recruited neurons in the landmark and azimuth matrix. A place cell learned in the location A responds maximally in A and creates a large decreasing field around A. Such a system is able to learn several regions of the environment. It can perform visual localization.

2.2 Place-movement association to define trajectories

Originally, the robot follows a random direction. When the robot takes a wrong direction, the human teacher can use a leash to drag the robot toward the desired path. When doing so, the user modifies the dynamics of the robot and the motor command of the wheels orientates the robot in the desired direction. The resulting change in proprioception (orientation change) triggers the learning of the association between the movement being done and the visual panorama (the current location). The orientations and the speeds are discretized so that each neuron S_i corresponds to a given orientation and a given speed. Their activities are calculated with (2).

$$s_i(t) = \sum_{p=1}^{n_{PC}} \omega_{pi}^{S}(t) \cdot PC_p(t)$$
$$S_i(t) = V(t) \cdot S_i^d(t) + (1 - V(t)) \cdot \left(\frac{s_i(t)}{s\max(t)}\right) \quad (2)$$

When the teacher modifies the dynamics of the robot, the change is detected and the vigilance signal

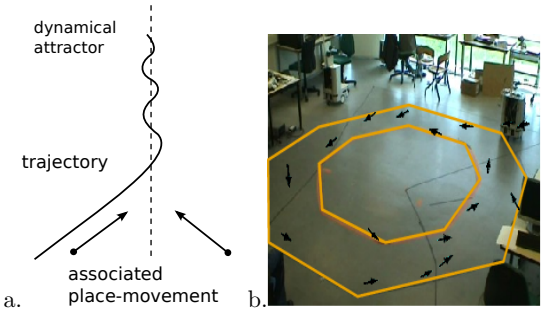


Figure 4: Learning of a path by correction of the learned dynamics. Each arrow represents a correction applied by the teacher. Hence, the robot learns the association between the place and its orientation. a.) A straight line trajectory can be defined by only two state-action associations. This trajectory is an attractor as from every point in the space, the robot will reach the line. b.) An example of trajectory that can be learned. After 3 rounds of learning, the robot is fully autonomous, the professor does not need to correct it anymore.

V becomes equal to 1. $s_{\max} = \max_{i=1..n_S}(s_i)$ is used for an output normalization with n_S the number of actions. The output S_i can either be the action predicted by the place cells or the desired action $S_i^d(t)$ (orientation and speed) that is determined from the action performed by the robot with or without the intervention of the teacher. The weight ω_{pi}^S of the connection between the p^{th} place cell and the i^{th} action is adapted according to a learning rate $\epsilon(t)$ and the learning rule (3) which is inspired of Widrow&Hoff gradient descent rule:

$$\frac{d\omega_{pi}^S}{dt} = (S_i^d(t) - s_i(t)) \cdot PC_p(t) \cdot V(t) \cdot \epsilon(t) \qquad (3)$$

Building sensorimotor attractors by associating movements to different regions of the environment ([Giovannangeli and Gaussier, 2010], Figure 4) enables the robot to reproduce the learned trajectory. A more sophisticated version of this associative learning exists which is not described in this paper [Giovannangeli and Gaussier, 2010]. In the case of the short learning of the navigation experiment described in Section 2.3, that version shows little difference with the presented version.

2.3 Application of the model to a U-shaped trajectory

The robot follows a direction which is corrected by the human teacher whenever it is too far from the desired trajectory. Modifications of the dynamics of the robot imply the learning of new place cells (Figure 5). The Neural Network model enables the reproduction of this simple trajectory after only three runs,

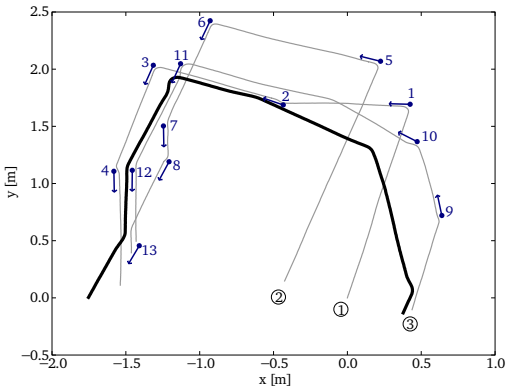


Figure 6: Experimental data corresponding to a simple U-shaped path. Top: trajectories in the Cartesian space (using odometry, reset at the beginning of each run). Grey lines are the trajectories during which learning occurred. The successive starting position are numbered. The reproduction (no correction) trajectory is in black. Dark blue arrows are the learned place-orientations.

using only the corrective interaction from the human teacher (Figure 6). During each run, the following data are recorded: activations of place cells (outputs from the Neural Network), the current position and orientation of the robot (using both odometry and magnetic compass). These data are used to train the GMM-based system for the navigational task.

3. Gaussian Mixture Model approaches for robot navigation

In this section we propose two uses of Gaussian Mixture Models in order to learn trajectories in navigation task. The first implementation is based on an ideal situation with relevant data directly available. Secondly, we propose an implementation which uses subjective visual cues in a more autonomous and realistic approach.

3.1 A direct transposition of the manipulation model using Cartesian coordinates

A Gaussian Mixture Model defines a probability density function on the state space of the robot.

$$p(\xi) = \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} e^{-\frac{1}{2}\left((\xi-\mu_k)^\top \Sigma_k^{-1}(\xi-\mu_k)\right)} \qquad (4)$$

D is the dimension of the state space and k is the number of states. π_k are prior probabilities, μ_k are mean matrices, Σ_k are covariance matrices and ξ are points of the state space. The GMM is characterized by the three parameters π_k, μ_k, Σ_k. Given

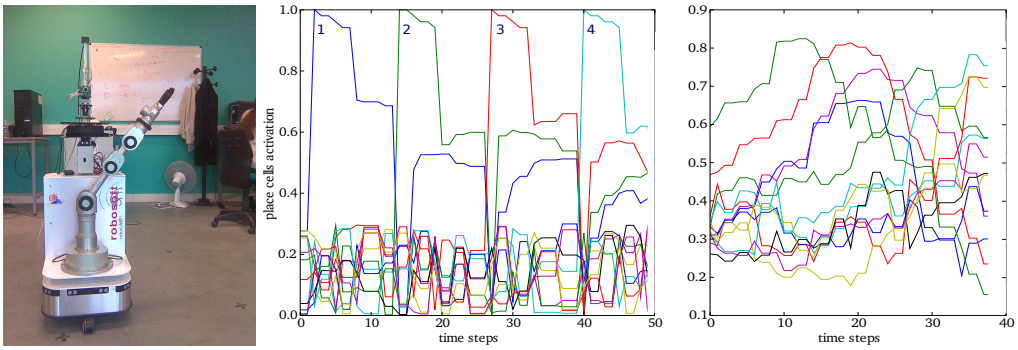


Figure 5: Experimental data corresponding to a simple U-shaped path. Left: Robulab mobile platform used for the experiments. Middle: place cell activation during trajectory while learning phase. Peaks correspond to the creation of new place cell, when user sets a new target azimuth (sharp turns in the Cartesian space) Right: place cell activation during the reproduction phase.

a training set of n datapoints (ξ_i), an Expectation-Maximization algorithm is used to find the parameters π_k, μ_k, Σ_k that maximizes the likelihood (6) of this training set. ***Expectation:*** We first estimate for each point of the training set the probability that this point ξ_i is generated by each Gaussian (or state k) $p(k|\xi_i)$.

$$p(k|\xi_i) = \frac{p(\xi_i|k)p(k)}{\sum_{j=1}^{K} p(\xi_i|j)p(j)} \tag{5}$$

where $p(\xi_i|k) = \mathcal{N}(\xi_i, \mu_k, \Sigma_k)$ and $p(k) = \pi_k$. The overall likelihood of observing the given training set with the given model parameters is:

$$\mathcal{L}(x) = \prod_{i=1}^{n} \left(\sum_{k=}^{K} p(\xi_i|k)\pi_k \right) \tag{6}$$

This is used as measure of convergence and performance of this algorithm. ***Maximization:*** Means and covariances of each Gaussian are recomputed by weighting each data point by the probability $p(k|\xi_i)$. These two steps are iterated until convergence.

The Gaussian Mixture Regression (GMR) enables to probabilistically complement partial information. Let $\xi = [\xi^O \xi^I]$ be a point in the state space with ξ^I that is known. The GMR enables to estimate ξ^O by taking the mean of the expectation distribution (7) (see [Cohn et al., 1996] for more details).

$$p(\xi^O|\xi^I) \sim \sum_{k=1}^{K} p(k|\xi^I)\; p(\xi^O|\xi^I, k), \tag{7}$$

We now consider that the state space is $\xi = [\theta\ v\ x\ y]$ with θ the absolute azimuth, v the linear velocity and x, y the Cartesian position of the robot. The data recorded during the three training runs of the NN system (see Section 2.3) are used to generate the training sets (ξ_i) for the Gaussian Mixture Model (GMM). The training is off-line. A simulation of a robot using the orientation commands retrieved from the GMR has been realized. The resulting trajectory with a 4 states GMM is shown in Figure 7. This simulation tells us that it makes sense to use such a continuous state action model to learn trajectories for a differential drive robot.

The drawback of this approach is that the absolute Cartesian position of the robot must be known, which in this case was obtained by the carefully recalibration of an odometer. This approach can be used only if the robot has access to the absolute Cartesian position. Odometry is not reliable for that purpose. Without a regular recalibration, the possible drift in the computation of the position can lead to an important inaccuracy. A robust localization system is necessary, such as a statistical localization system (Kalman Filter, Particle filter [Thrun et al., 2001]), or an external visual tracking system with accurate calibration. These methods are still costly to settle, since they require either extra computation, or extra hardware.

3.2 Navigation with GMM training using place-cell activity

In a more developmental and autonomous approach, the robot should rely on subjective visual cues to localize itself. This would allow more robustness and adaptation as the robot would not need a predetermined map of the environment to localize itself. The raw landmark-azimuth pairs introduced in the neural approach in Section 2.1 could be used. Though, as the dimension of the inputs increases, the computational time for the GMM model increases like $O(n^3)$. Moreover, the structure of the visual information with noise and occulting implies that the number of pertinent information may not be stabi-

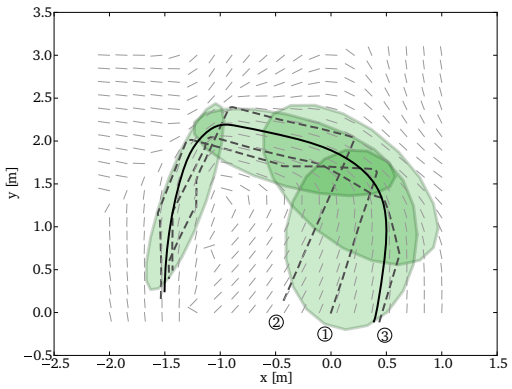


Figure 7: Generated trajectory (black), and model using information on the absolute Cartesian position ($p(\theta, v|x, y)$). Dashed lines are demonstration data, GMM states are represented with green ellipses and the light gray field lines represent target azimuths at given position.

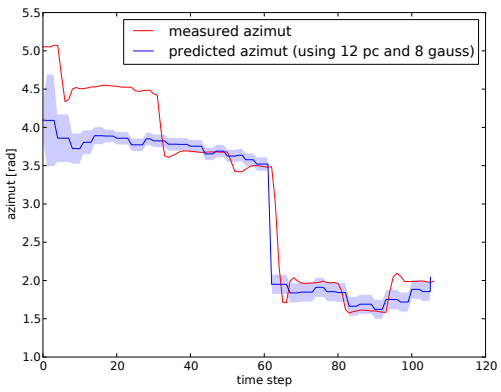


Figure 8: Predicted azimuths using place cell activation based on Equation 8 and using the data from 12 place cells. The surface in light blue represent the incertitude on the target angle given by the probability density ($\pm\sigma$).

lized which makes the training more difficult.

Place cells offer an efficient pre-processing of the visual inputs for the training. In the Neural Network model, they have proved to be robust to noise and occultation (keeping the same ordering), and they represent a smaller amount of information. The module that calculates the place-cell activities is added to the Gaussian Mixture Model system so that the state-space is now defined as $\xi = [\theta(t)\ PC(t)]$ with $PC(t) = (PC_1(t), \cdots, PC_N(t))$. The number of place cells is chosen arbitrarily. At every time step we can retrieve a continuous value for the target azimuth using (7) which becomes equation (8). It defines a continuous and probabilistic mapping between sensory inputs $PC(t)$ and the motor command $\theta(t)$.

$$\theta(t) \rightarrow p(\theta|PC(t)) \tag{8}$$

The three training runs realized using the Neural Network system (Section 2.3) provide the training data. At the end of these runs, twelve place cells had been learned. The GM Model is trained using 8 Gaussians and these 12 place cells. A Gaussian noise of variance 0.1 is added to place cell activations during the training to simulate noisy visual inputs. Place-cell activities from the test set with the Neural Network model are given to the GMM to predict azimuth target. In Figure 8, a comparison between this azimuth and the real one from the test run with the Neural Network system shows that using place-cell activations is acceptable for retrieving a target azimuth from the GMM. Because place cells are generated during turns, there are no place cells at the beginning of trajectories. That can explain why the model performs badly at the beginning.

The previous experiment allows us to compare the retrieved orientation command on a test trajectory. We now simulate place cells by Gaussian activations. The parameters of the Gaussian activations of the place cells can be estimated from the recorded data provided by the learning and test runs for the Neural Network system. It is then possible to associate place-cell activities to every position in the environment. These activities are given to a simulated robot to reproduce the learned trajectory. Three extrapolated place cells are used. The different parts of the trajectory correspond to different dominant states of the GMM. It appears that the states defines a new topology that is built on top of the place cells. The result of the simulation is given in Figure 9. Using such an approximation shows that the GMM based system is well able to reproduced the desired trajectory by using radial basis activation for place cells.

4. Encoding and learning strategies

Both approaches succeed in solving the task. However they present differences in encoding that influences the abilities of the system. The Neural Network system is based on vision. It gathers information (azimuth, landmarks) into place cells which are associated to desired orientations when the robot trajectory is corrected. The Gaussian Mixture Model encodes statistically pertinent information. Using Cartesian coordinates as inputs is efficient for navigation, but place cells can also be used.

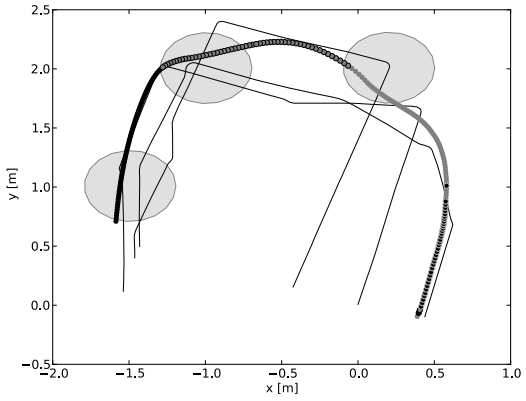


Figure 9: Simulated trajectory with a GMM system using place-cell activities approximated by Gaussians. The light gray ellipes display simulated place cells. Black thin lines are training trajectories. The reproduction trajectory is represented by successive points. For each point, the most probable Gaussian among the four in the Gaussian Mixture Model gives its design to the point. There are four different designs, one for each Gaussian state.

4.1 Optimized encoding and long time learning

With several runs and corrections whenever the NN system is not showing the required behavior, a dynamical attractor can be constructed (Figure 4). Each time the trajectory of the robot is corrected, another place cell is learned. But there is no guarantee that the new place cell will only correct the mistake. As place cells have a wide recognition activity, new place cells can interfere with older ones. The attractor stability and generalization rely on the number and the positions of place cells recruited to learn the actions [Giovannangeli and Gaussier, 2010]. Several place cells can be needed to stabilize the dynamics over the desired path, especially under strong constraints (if small variance on the resulting trajectory is needed). With many place cells, there can be a redundancy in the encoded information. A specific implementation for adaptive orientation cells can enhance the model [Giovannangeli and Gaussier, 2010]. Before recruiting new place cells, it is possible to try to adapt the orientation associated to an already known cell. However, the optimized repositioning or pruning of place cells is currently an unsolved problem. As learning continues, with changes of the environment, more and more place cells may be recruited leading to an oversampling of the space.

Generalizing from several samples enables the GMM system to optimize the encoding of the trajectory by computing Gaussian states that minimize redundancy and maximize available information. There is no risk of bad interference as every state participates during learning. With the tested implementation, the number of Gaussians that will be used by the system must be given in advance. Other specific implementations can solve this problem and enable incremental learning [Calinon et al., 2009]. As learning continues, the first datasets may become obsolete. If the environment changes too much, the system requires a complete re-learning from new correct data.

4.2 Interaction

The presented Neural Network implements on-line learning by correction of the behavior. The robot can reproduce immediately a learned walk, even if it may not exactly be the one that is shown. As the teacher can directly see what has been learned by the robot, bad orientation can be instantaneously corrected. This new association immediately modifies the dynamics of the robot which can be corrected again, etc. This incremental approach enables the system to focus on ill-learned part of the trajectory.

The learning of the GMM system can be qualified of slow as the robot must first be driven completely passively before it can do anything. The whole trajectory must be shown several times so that the robot get enough training data. The learning is off-line and requests several samples of data so that statistics about the task can be built. Each sample requires the teacher to realize a trajectory from start to end so they are tedious for the human teacher. The encoding depends on the complete training. What was learned by the robot can not be known before the end of the training. If the robot must be corrected, another set of demonstrations by the human teacher is necessary to retrain the system.

4.3 Quality of the learned trajectory

The trajectory generated using GMM-based system is a smoother approximation of the U-shaped desired trajectory than the one generated with the Neural Network system. In the Neural Network system, a possible solution to avoid interferences is to use discretized orientations and a competition between the place cells. When using a strict competition, the trajectories often appear as straight concatenated lines, and do not seem very natural. A soft competition over the recognized place cells suppress this problem (a few winner place cells activate a mixture of associated directions allowing smoother trajectories). In the GMM system, since both inputs and outputs are continuous, when the system is between two known positions, it automatically combine the required behaviors yielding nicer trajectories. Because the actions are demonstrated by a human, the desired trajectory is hidden by variations. The statistical analysis of the datasets enables the GMM to

retrieve the optimal trajectory and it can even determine the constraints and degrees of freedom for a given walk. According to the variance of the expectation distribution, the trajectory can be more or less constrained.

4.4 Explicit and implicit encoding, fast and slow learning

The difference in encoding can be summarized by explicit versus implicit encoding. The NN system directly encodes the state-action associations that are perceived during learning. The GMM system distributes the implicit encoding between different states. No state is specifically related to an event. This difference of encoding is bound to a difference in learning. Explicit associations can be learned rapidly and independently like corrections in the NN system. On the contrary, implicit associations require several data. The learning is slowed down by the need of several passive demonstration to create several training datasets. The system can not reproduce any action until the end of this training.

5. Discussion

The context of this study is learning in a situation of interaction. More specifically, we focused on two learning by demonstration approaches that are based on state-action associations. We do not tackle reinforcement learning as it does not really correspond to demonstration by a human teacher. The robot can not explore its environment to build reinforcement evaluation. It must use the information provided by the interaction. Linking interaction and reinforcement is an ongoing work (see [Hirel et al., unpublished]).

In [Zukowgoldring and Arbib, 2007], the authors defend the hypothesis that the main feature of interaction between an infant and its caregiver is not the direct transfer of knowledge but the reduction of the research space in order to help the infant to learn new tasks. The fast corrective learning in the Neural Network system corresponds quite well to this definition. As the lively space is reduced, the robot is led to reproduce what the human teacher wants it to do with more or less accuracy. But, this is limited. Rather than increasing more and more the number of corrective rules that define the action, a statistical analysis of the structure of the possible actions could optimize the encoding. In the case of a robot, the GMM system could complement the Neural Network system to enhance the learning abilities.

This reflexion driven by a developmental point of view is also based on an anatomical analysis. [Eichenbaum et al., 1994] stresses the complementarity of two distinct structures in the brain that are the Neocortex and the Hippocampus. These structures have specificities which are very similar to the specificities of the two encoding and learning strategies that we study in this paper. The Hippocampus is an explicit associative memory that can acquire rapidly information. As there is little inference from the encoded items, the interferences are limited. It is also related to novelty detection like changes of orientation in the case of our robot. This structure corresponds well to the place-cell associations learned in the Neural Network architecture. The Neocortex is an implicit memory as it does not directly encode specific events. It can discover gradually the statistical structures of experiences, by accumulating learning and data. This structure corresponds to the GMM process. Motor and PreMotor Cortex can be the location where the invariant features of movements and trajectories are retained. Some substructures and other structures of the brain participates in specific manners in the cognitive process. The Entorhinal Cortex interfaces the hippocampus and the neocortex. In [Arleo and Gerstner, 2000], Entorhinal Cortex is the location of the place cells. Place-cell activities can be used by both structures and enable exchanges between Hippocampus and Neocortex. This corresponds to the system we simulated in Section 3.2. The cerebellum provides interpolation and prediction abilities to the brain system. With interpolation and conditioning, it can improve the quality of movements. There exist a transfer of knowledge between the hippocampal episodic short term memory and the neocortical long term memory [McClelland et al., 1995]. This happens mainly during sleeping time. The cerebellum can play an important role in this process. Its predictive capacities can be used for internal rehearsal of episodic memories so that Neocortex can learn implicit representation. This long term representation can then be used by the Striatum to generate routine movements. During the transfer between short term and long term memory, a change in representation, ie. of encoding, can occur. This process is called memory consolidation. It has been observed in animals and in human [McClelland et al., 1995]. In [Kulić and Nakamura, 2009], memory consolidation is used with incremental learning so that the system can learn on-line with additive stability coming from consolidation memory which occurred both on-line during wake time and off-line during equivalent sleeping time. This consolidation enabled a better categorization of the action. This memory consolidation has also proved to make human-robot collaboration easier in [Ogata et al., 2004].

In this paper, the NN system and the GMM system have been studied separately. As a future work, we suggest that they are gathered in a whole architecture that benefits from the complementarity of the two strategies. Once the robot has acquired a new

rough behavior, it can reproduce the trajectory over and over providing the necessary datasets for GMM training. Learning can be done in two times: a first rough learning of the task which is then refined using more data to determine the invariant features of the actions that solves the task. One model can take over the second one according to the situation. When facing already known situations, the GMM based system can produce an optimized adapted behavior and when facing new situations, the Neural Network enables interactive corrections to incrementally generate an adapted behavior.

The discussion provided here has been based on a navigation experiment, but the conclusion should be extended to action selection problem in general. We believe that the same explicit/implicit memory structures can be involved not only in navigation, but also in manipulation tasks or even higher complex tasks mixing different kinds of behaviors. Future works will study this hypothesis. As it can be seen in Section 4, the conclusions drawn may not be restricted to the algorithms studied in this paper. The conclusions should be extended to algorithms that could be classified either as fast learning, explicit encoding or as slow learning, implicit encoding.

In conclusion, the two strategies studied in this paper must be considered as two complementary encoding and learning strategies. One can not replace the other one, it is not a matter of trade-off between the two strategies. According to how evolution built our brain, both strategies must be present in order to provide the cognitive system with most efficiency as considering reactivity and interactivity during learning with good inferences to optimally retain any demonstrated knowledge.

6. Acknowledgement

This work was supported by the French Region Ile de France, the Institut Universitaire de France (IUF), the FEELIX GROWING european project and the INTERACT french project referenced ANR-09-CORD-014.

References

Arleo, A. and Gerstner, W. (2000). Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological Cybernetics*, 83(3):287–299–299.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Survey: Robot Programming by Demonstration. In *Handbook of Robotics*, volume chapter 59. MIT Press.

Calinon, S., Caldwell, D., Billard, A., and D'halluin, F. (2009). A probabilistic approach to learn and reproduce dynamics of human motion by imitation. In *Proceedings of 2009 IEEE International Conference on Humanoid Robots*.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *CoRR*, cs.AI/9603104.

Eichenbaum, H., Otto, T., and Cohen, N. J. (1994). Two functional components of the hippocampal memory system. *Behavioral and Brain Sciences*, 17(03):449–472.

Giovannangeli, C. and Gaussier, P. (2010). Interactive teaching for vision-based mobile robots: A sensory-motor approach. *IEEE Transactions on Man, Systems and Cybernetics, Part A: Systems and humans*, 40(1):13–28.

Giovannangeli, C., Gaussier, P., and Désilles, G. (2006). Robust mapless outdoor vision-based navigation. In *IEEE/RSJ International Conference on Intelligent Robots and systems*, Beijing, China. IEEE.

Kulić, D. and Nakamura, Y. (2009). Incremental learning and memory consolidation of whole body human motion primitives. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 17(6):484–507.

Lagarde, M., Andry, P., Gaussier, P., Boucenna, S., and Hafemeister, L. (2010). *Proprioception and Imitation: On the Road to Agent Individuation*, volume 264 of *Studies in Computational Intelligence*, pages 43–63. Springer Berlin / Heildelberg.

McClelland, J. L., Mcnaughton, B. L., and O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychol Rev*, 102(3):419–457.

Ogata, T., Sugano, S., and Tani, J. (2004). Open-end human robot interaction from the dynamical systems perspective: Mutual adaptation and incremental learning. In *Innovations in Applied Artificial Intelligence*, pages 435–444. Springer Berlin / Heidelberg.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141.

Zukowgoldring, P. and Arbib, M. (2007). Affordances, effectivities, and assisted imitation: Caregivers and the directing of attention. *Neurocomputing*, 70(13-15):2181–2193.