

# Apprentissage de comportements pour un robot autonome

## THÈSE

présentée et soutenue publiquement le 19 Décembre 2002

pour l'obtention du

**Doctorat de l'université Pierre et Marie Curie – Paris 6**

(spécialité informatique)

par

**Louis Hugues**

### Composition du jury

*Rapporteurs :* **Philippe Gaussier**, Professeur (Université Cergy)  
**Jean Sallantin**, Directeur de Recherche (LIRMM - CNRS)

*Examineurs :* **Rachid Alami**, Directeur de Recherche (LAAS - CNRS)  
**Patrick Gallinari**, Professeur (Université Paris 6)  
**Jean-Daniel Zucker**, Professeur (Université Paris 13)

*Directeur :* **Alexis Drogoul**, Professeur (Université Paris 6)

Mis en page avec la classe thloria.

## Résumé

Les robots autonomes et mobiles devront, pour nous être utiles, effectuer des tâches dans des environnements réels tels que les hôpitaux, les entreprises, les lieux publics ou encore les domiciles des particuliers. Ces robots devront se comporter conformément à l'attente des utilisateurs humains, dans des environnements difficilement modélisables et en s'appuyant sur des perceptions incomplètes, imprécises et bruitées. Dans ces conditions, très différentes de celles du simulateur logiciel ou de l'environnement préparé, la question se pose de savoir comment les robots peuvent acquérir les comportements nécessaires. C'est cette transmission de compétences à un robot mobile en environnement réel qui forme la problématique centrale de cette thèse. Cette dernière repose sur l'hypothèse principale selon laquelle l'apprentissage des comportements peut résulter des interactions entre le robot et son tuteur humain dans l'environnement de destination. Pour répondre à ce problème, un nouveau modèle d'apprentissage est présenté, capable de capter ces interactions en l'absence de connaissances à priori sur l'environnement et sur les tâches à réaliser. Le modèle d'apprentissage MPL (Micro Population for Learning) est conçu pour produire des comportements synthétiques à partir de démonstrations réalisées par un tuteur humain. Pour ce faire, MPL repose sur une représentation distribuée constituée par une population de cellules élémentaires non hiérarchisées; après apprentissage ces cellules déterminent collectivement les actions à effectuer et sont capables de prendre en compte les différentes formes de bruits présentes dans l'environnement. La phase d'expérimentations démontre la faisabilité de l'apprentissage en situation réelle par cette approche distribuée. Elle est appliquée à des comportements situés correspondant à des tâches élémentaires, fréquemment réalisées par un robot, mais difficiles voire impossibles à programmer. Les performances, mesurées en terme de rapidité d'apprentissage et de qualité de réalisation des comportements, s'avèrent meilleures pour ce types de problèmes que celles des méthodes plus classiques telles que les réseaux de neurones (PMC). Le modèle est ensuite utilisé dans un cadre interactif dans lequel le tuteur amende et modifie en temps réel un comportement appris. Cette dernière partie ouvre la voie vers un façonnage interactif et social des aptitudes des robots.

**Mots-clés:** Robotique autonome, Apprentissage interactif, Perception, Comportements situés

## Abstract

To be useful to us, the autonomous mobile robots will have to perform tasks in real environments such as hospitals, offices, factories or personal homes. Those robots will have to behave in ways that correspond to the users expectations, in environments which are difficult to model, and supported by partial, imprecise and noisy perceptions. In those conditions, which are quite different of those of the software simulator or of the prepared environment, the question is therefore to know how the robots can acquire the requested behaviors. This question of the acquisition of skills in real environments constitute the central problematic of this thesis. I make the main hypothesis that the appropriate learning of behavior can result of interactions between the robot and its human tutor, in the destination environment. To answer to this problem I present a new learning model, able to capture those interactions without a priori knowledge about the environment or the task to learn. The learning model MPL (Micro Population for Learning) is conceived to produce synthetic behaviors from demonstrations realized by a human tutor. To do so, MPL uses a distributed representation made of a population of cells; after a learning phase, those cells determine collectively the appropriate actions and are able to deal with the various forms of noise which are present in the environment. The experimentations phase demonstrates the feasibility of the learning in real situations with this distributed approach. It is applied to situated behaviors corresponding to elementary tasks, frequently performed by a robot, but difficult and somehow impossible to program. The performances, which are measured in terms of learning speed and quality of behavior realization, appear to be much better for this kind of problems than the classical methods such as neural networks. The model is then used in an interactive framework where the tutor amends and modifies in real time a previously learned behavior. This last part prepares the way toward an interactive and social shaping of robots aptitudes.

**Keywords:** Autonomous robots, Interactive learning, Perception, Situated behaviors

# Table des matières

<b>Chapitre 1</b>	
<b>Introduction générale</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Apprendre des comportements d’usages de l’environnement . . . . .	3
1.2.1 Un exemple de comportement d’usage de l’environnement . . . . .	4
1.3 Le projet <i>MICRobES</i> . . . . .	4
1.4 Organisation de ce document . . . . .	5

---

---

## Partie I Les difficultés de l’acquisition de comportements en environnement réel

---

---

<b>Chapitre 2</b>	
<b>Les difficultés propres aux perceptions</b>	
2.1 Introduction . . . . .	11
2.2 Les sens des robots . . . . .	13
2.3 Altération des perceptions . . . . .	16
2.3.1 Exemples réels . . . . .	21
2.4 Difficulté d’associer objet et perception . . . . .	24
2.4.1 Que faut-il percevoir ? . . . . .	27

**Chapitre 3**

**Les difficultés propres à l’environnement**

3.1	Introduction . . . . .	29
3.2	Caractérisation classique de l’environnement . . . . .	31
3.3	Caractéristiques supplémentaires de l’environnement . . . . .	33
3.4	Percevoir et agir dans l’environnement . . . . .	35
3.5	Deux conséquences essentielles . . . . .	38
3.5.1	Les robots réels et leurs environnements ne sont pas modélisables .	38
3.5.2	Les robots réels et leurs environnements ne sont pas simulables . .	39

**Chapitre 4**

**Les difficultés propres à l’acquisition de comportements**

4.1	Introduction . . . . .	41
4.2	La programmation des comportements . . . . .	42
4.3	L’acquisition par l’apprentissage . . . . .	43
4.3.1	Transmission téléologique des intentions . . . . .	43
4.3.2	Transmission empirique des intentions . . . . .	45
4.4	Biais . . . . .	45
4.5	Education . . . . .	46
4.5.1	Durée . . . . .	47
4.5.2	Intégration . . . . .	47
4.6	Conclusion . . . . .	48

**Chapitre 5**

**Incidences sur les systèmes d’acquisition de comportements**

5.1	Introduction . . . . .	49
5.2	Propriétés requises . . . . .	50
5.3	Incidence sur les moyens utilisables . . . . .	51
5.4	Principale hypothèse . . . . .	52

**Chapitre 6**

**Approches empiriques et interactives de l’apprentissage de comportements**

6.1	Introduction . . . . .	53
6.2	Description de systèmes empiriques . . . . .	57
6.2.1	D.A. Pomerlau - apprentissage supervisé numérique. . . . .	57
6.2.2	G.Atkeson and S.Schaal - Apprentissage par démonstration . . . .	58

6.2.3	J.Hayes and J Demiris - Apprentissage par Imitation . . . . .	59
6.2.4	M.N. Nicolescu, M. Mataric : Apprentissage de tâches . . . . .	60
6.3	Conclusion . . . . .	60

---



---

## Partie II Un modèle pour l'acquisition de comportements

---



---

### Chapitre 7

#### Elaboration d'un modèle d'acquisition de comportements

7.1	Introduction . . . . .	67
7.2	Pistes pour construire le modèle . . . . .	67
7.2.1	Affordances . . . . .	68
7.2.2	Population . . . . .	69
7.2.3	Perceptions élémentaires . . . . .	70
7.3	Hypothèses structurant le modèle . . . . .	71
7.3.1	Hypothèse de distribution . . . . .	72
7.3.2	Hypothèse de Tesselation (micro-perception) . . . . .	72
7.3.3	Hypothèse du couplage perception/action et de concomitance . . . . .	73
7.3.4	Hypothèse du façonnage (shaping) . . . . .	73

### Chapitre 8

#### Cadre pour le modèle d'acquisition de comportements

8.1	Introduction . . . . .	75
8.2	Le robot élémentaire (RE) . . . . .	75
8.3	Apprentissage à partir de démonstrations . . . . .	78
8.4	L'acquisition dans le cadre de l'apprentissage supervisé . . . . .	80
8.4.1	Expérimentation préliminaire . . . . .	81
8.5	Schéma général du processus de synthèse . . . . .	84

**Chapitre 9****MPL Modèle pour la synthèse supervisée de comportements**

9.1	Introduction . . . . .	87
9.2	Structure du modèle <b>MPL</b> . . . . .	88
9.2.1	Champ perceptuel et tesselles . . . . .	89
9.2.2	Cellules . . . . .	91
9.2.3	Mécanisme de synthèse . . . . .	93
9.2.4	Mécanisme de réalisation autonome . . . . .	94
9.2.5	Schéma général . . . . .	94

**Chapitre 10****MPL-t Modèle taxique**

10.1	Introduction . . . . .	97
10.2	Mécanisme de synthèse . . . . .	100
10.2.1	Selection des cellules : <i>generate_pop</i> . . . . .	100
10.2.2	Détermination de l'action : <i>inform_pop</i> . . . . .	100
10.2.3	Détermination du gain . . . . .	101
10.3	Mécanisme de réalisation autonome . . . . .	102
10.4	Application à un exemple élémentaire . . . . .	102
10.5	Complexité . . . . .	105
10.6	Remarques . . . . .	105

**Chapitre 11****MPL-s Modèle à échantillons**

11.1	Introduction . . . . .	109
11.2	Sélectivité de percepts à n pixels pris aléatoirement . . . . .	109
11.3	Nouvelle structure de la cellule . . . . .	111
11.4	Mécanisme de synthèse . . . . .	112
11.5	Mécanisme de réalisation autonome . . . . .	114
11.6	Remarques . . . . .	114

**Chapitre 12****Compléments au modèle d'acquisition**

12.1	Introduction . . . . .	115
12.2	Mesure d'utilité . . . . .	115
12.3	Perception active . . . . .	116
12.4	Sélection de comportements . . . . .	117

---

12.5 Conclusion . . . . .	119
---------------------------	-----

<b>Chapitre 13</b>
--------------------

<b>Structuration du champ perceptuel</b>
--

13.1 Introduction . . . . .	121
13.2 Quelles propriétés perceptuelles ? . . . . .	123
13.3 Constitution des plans perceptuels . . . . .	126
13.3.1 Filtres . . . . .	126
13.3.2 Classeurs . . . . .	127
13.3.3 Conclusion . . . . .	131

---

---

## Partie III Expérimentations

---

---

<b>Chapitre 14</b>
--------------------

<b>Plate-forme expérimentale</b>
----------------------------------

14.1 Introduction . . . . .	135
14.2 Plate-forme robotique . . . . .	135
14.2.1 Senseurs . . . . .	136
14.2.2 Effecteurs . . . . .	136
14.2.3 Moyens informatiques embarqués . . . . .	137
14.3 Implémentation du robot élémentaire (RE) . . . . .	138
14.4 Plate-forme d'apprentissage . . . . .	139
14.4.1 Fichiers de démonstration . . . . .	140
14.4.2 Pilotage à distance . . . . .	140
14.4.3 Interface utilisateur . . . . .	140
14.5 Mise en oeuvre logicielle . . . . .	142
14.5.1 Environnement logiciel . . . . .	143
14.5.2 Composants et architecture dynamique . . . . .	143

**Chapitre 15**

**Acquisition de comportements**

15.1	Introduction . . . . .	147
15.2	Expérience GOSTOP . . . . .	147
15.2.1	Démonstrations . . . . .	148
15.2.2	Résultats . . . . .	148
15.2.3	Configuration . . . . .	149
15.3	Expérience SLALOM . . . . .	151
15.3.1	Démonstrations . . . . .	151
15.3.2	Résultats . . . . .	152
15.3.3	Comparaison avec un réseau de neurones PMC . . . . .	154
15.4	Expérience DOCKING . . . . .	158
15.4.1	Démonstrations . . . . .	159
15.4.2	Résultats . . . . .	160
15.5	Expérience FUSION . . . . .	162
15.5.1	Démonstrations . . . . .	162
15.5.2	Résultats . . . . .	162
15.6	Expérience OCCLUSION . . . . .	165
15.6.1	Résultats . . . . .	166
15.7	Conclusion . . . . .	166

**Chapitre 16**

**Propriétés du modèle MPL**

16.1	Introduction . . . . .	169
16.2	Retour sur les propriétés attendues . . . . .	169
16.3	Limites structurelle de <b>MPL</b> . . . . .	171
16.4	Principe inductif . . . . .	172
16.4.1	Combinaison de décisions . . . . .	172
16.4.2	Compression des démonstrations . . . . .	173
16.4.3	Biais structurel . . . . .	173
16.5	Conclusion . . . . .	174

---



---

**Partie IV Apprentissage Interactif et Perspectives**

---



---

---

**Chapitre 17****Vers l'apprentissage interactif**

17.1	Introduction . . . . .	177
17.2	Principes . . . . .	178
17.2.1	Amorçage et adaptation . . . . .	178
17.2.2	Opérateurs élémentaires d'adaptation . . . . .	178
17.2.3	Etablissement d'un processus éducatif interactif . . . . .	180
17.3	Etude expérimentale . . . . .	181
17.3.1	Les modalités d'adaptation mises en oeuvre . . . . .	181
17.3.2	Modalité d'ajustement . . . . .	181
17.3.3	Modalité d'ajout d'information . . . . .	182
17.4	Expérimentations et résultats . . . . .	183
17.4.1	Reprise de l'expérience SLALOM . . . . .	183
17.5	Conclusion . . . . .	184

**Chapitre 18****Perspectives et conclusion**

18.1	Extensions de <b>MPL</b> . . . . .	187
18.1.1	Extension du champ perceptuel . . . . .	187
18.1.2	L'articulation temporelle de l'action . . . . .	188
18.1.3	L'apprentissage interactif . . . . .	188
18.2	Abstraction par interaction et réorganisation . . . . .	188
18.3	Conclusion Générale . . . . .	190

**Bibliographie****193**



# Chapitre 1

## Introduction générale

### 1.1 Introduction

La conception d'un être artificiel est depuis longtemps une préoccupation des humains tant elle est conçue à la fois comme une voie pour la compréhension de notre propre fonctionnement et comme un moyen de nous décharger de tâches pénibles ou inaccessibles. Depuis Talos le géant de bronze qui protégeait l'île de Crète en passant par le canard de Vaucanson et le joueur d'échecs du Baron Kempelen jusqu'aux robots d'Asimov ou de Star-Wars, nombre de tentatives effectives ou imaginaires se sont succédées pour tenter de doter des machines d'un fonctionnement autonome. Chacune d'elles s'est établie entre la volonté d'élucider le mystère du fonctionnement autonome des êtres vivants et celle de produire des artefacts extra-ordinaires et utiles.

L'avènement de la cybernétique dans les années 40 au travers des travaux de Norbert Wiener puis celles des sciences et techniques de l'information ont fourni un nouveau support plus adapté pour la conception de ces entités autonomes. Les tortues électroniques de Greg Walter capables de se recharger et de d'éviter les obstacles en n'utilisant que des principes analogiques furent en 1950 l'un des premiers exemples de véhicules autonomes<sup>1</sup>. Elles furent suivies dans les années 60 par les premiers robots utilisant un ordinateur numérique associé à des senseurs et effecteurs ; ce fut le cas du robot Shakey de l'université de Stanford qui fut principalement conçu pour l'étude du raisonnement automatique et de la planification dans

---

<sup>1</sup>Précédées en 1890 par Nikola Tesla concepteur d'un véhicule aquatique radio commandé.

le cadre robotique (STRIPS) [Nilsson, 1984]. Vinrent ensuite les travaux de H. Moravec à Carnegie Mellon avec le robot Rover [Moravec, 1982], déjà plus orienté vers les problèmes de perception et de représentations appropriées de l'environnement. A partir des années 80, en conséquence des progrès technologiques et de la baisse des coûts, la robotique autonome s'est répandue dans de nombreux laboratoires donnant naissance à toutes sortes de robots à roues, à pattes, humanoïdes ou serpentiforme jusqu'à arriver en ces années 2000 vers des robots proposés à l'entreprise ou au grand public (Sony, Lego, Honda, IRobot, etc.). L'importance des budgets privés et publics consacrés par les deux principaux acteurs de ce domaine que sont les Etats-Unis et le Japon <sup>2</sup> <sup>3</sup> montrent que l'utopie de créer des entités artificielles autonomes n'en est en fait plus vraiment une.

Désormais, on peut s'attendre dans un futur relativement proche à ce que les robots autonomes fassent leur apparition dans des lieux aussi divers que les hôpitaux, les usines, les locaux d'entreprises ou les appartements. On s'imagine les voir rendre des services pour des tâches quotidiennes ou effectuer des opérations qui nous sont inaccessibles. Ainsi de nombreuses activités paraissent pouvoir être prises en compte par des machines robotiques telles qu'apporter le courrier, alimenter des photocopieuses, participer à une activité ludique ou éducative, vérifier des dispositifs techniques, surveiller des espaces stratégiques etc. Sans prendre la forme de robots de science-fiction, on verra sans doute les robots autonomes s'incarner dans des objets aussi divers que des véhicules autonomes, des aspirateurs, des fauteuils pour handicapés, des jouets mécaniques, des sous-marins nettoyeurs et pourquoi pas des nano-machines assistant des interventions médicales. Cette arrivée annoncée régulièrement dans les médias, se concrétise aujourd'hui au travers d'un nombre important de compétitions robotiques telles que RoboCup, Robot Cleaning Contest, Urban Search And Rescue et l'apparition de nombreuses entreprises. Elle s'appuie sur les progrès accomplis ces dix dernières années sur les senseurs permettant leur miniaturisation et une baisse significative des coûts<sup>4</sup>.

---

<sup>2</sup>Revue de l'Association française des Sciences et Technologie de l'information (ASTI), n° 82, sept. 2002.

<sup>3</sup>Revue La Recherche n° 350 Spécial Robots.

<sup>4</sup>Par exemple un robot insecte volant d'une dizaine de grammes doté de capacités visuelles élémentaires (Micromechanical Flying Insect) est en cours de développement à l'université de Berkeley [Yan *et al.*, 2001].

## 1.2 Apprendre des comportements d'usages de l'environnement

Aujourd'hui il est possible de faire naviguer des robots dans des environnements partiellement connus [Thrun *et al.*, 1999], de programmer et d'assembler facilement certains de leurs comportements [Arkin, 1998], de planifier leurs actions dans des environnements contrôlés [Beetz and McDermott, 1994; Alami *et al.*, 1998], de 'communiquer' dans une certaine mesure avec eux [Billard *et al.*, 1998; Breazeal and Scassellati, 1999]. Cependant il reste encore beaucoup à faire pour amener les robots dans notre quotidien. En particulier, leurs facultés d'adaptation, d'apprentissage et d'interaction avec les humains doivent être encore étendues et envisagées non pas dans des conditions contrôlées de simulateur ou de d'environnements de laboratoire mais bien dans les environnements que nous mêmes utilisons tous les jours. Comme l'indique Malik Ghallab (LAAS/CNRS) l'un des axes importants pour la robotique aujourd'hui est *"le robot personnel, serviteur accompagnant l'utilisateur sur son lieu de travail, avec une grande versatilité des environnements et des tâches; cela pose notamment de difficiles problèmes d'apprentissage, car l'utilisateur doit pouvoir "former" le robot aussi facilement qu'on forme aujourd'hui un apprenti"* (Revue ASTI n° 87, oct 2002).

La problématique traitée dans cette thèse est celle de la transmission de compétences à un robot mobile et s'inscrit précisément dans cette orientation. La gamme des compétences envisageables étant de loin trop grande pour être traitée dans une seule thèse, je me concentre ici sur **l'acquisition de comportements d'usage de l'environnement par un robot mobile**. J'entends par comportements d'usage ceux dont le robot a besoin à tout moment pour faire usage localement de son environnement : effectuer une manoeuvre difficile, utiliser un dispositif, négocier un passage, fuir un lieu interdit. Ce sont bien ces composants comportementaux dont la succession forme le quotidien du robot mobile et, comme on le verra dès la première partie ce sont ceux-là qui sont très difficiles à spécifier explicitement dès lors que l'on se trouve dans un environnement réel. On peut imaginer les combiner, les séquencer ou les planifier ultérieurement, mais la question à laquelle j'essaye de répondre ici est : comment les transmet-on ?

La question de l'acquisition de comportements d'usage de l'environnement est intéressante à plusieurs titres : il est d'abord utile de doter les robots de compétences concrètes utilisables

dans des scénarios réalistes ; ensuite, et plus généralement, cette approche donne un éclairage particulier sur les possibilités données aux machines pour appréhender un environnement réel non idéalisé.

### 1.2.1 Un exemple de comportement d'usage de l'environnement

On retrouvera à plusieurs reprises dans ce document un même exemple de comportement d'usage de l'environnement qui servira de fil rouge et d'illustration afin de conserver un caractère concret à l'exposé. Ce *comportement d'accostage* consiste pour un robot à s'approcher d'une station de recharge et à effectuer une manoeuvre d'arrimage afin de recharger ses batteries (fig. 1.1). On peut alors imaginer le scénario suivant : un nouveau modèle de station de recharge vient d'être installé dans une entreprise et il faut rapidement disposer d'un comportement pour permettre aux robots dont elle dispose d'en faire usage. Les questions qui découlent de ce scénario sont alors les suivantes : quels sont les moyens qui vont permettre l'acquisition de ce comportement ? Par qui seront-ils accessibles (experts ou utilisateurs) ? Pourra-t-on utiliser ces moyens pour acquérir d'autres comportements ? Il s'agit donc de trouver les moyens d'associer au symbole abstrait "Accostage" l'ensemble approprié de perceptions et d'actions *ancrées* dans l'environnement réel.



FIG. 1.1: *Eléments pour un comportement d'accostage : le robot considéré (à gauche), le dispositif d'arrimage (au centre), la station de recharge (à droite)*

La station de recharge que j'utiliserai dans ce scénario ainsi le dispositif d'arrimage, ont été conçus par François Sempé et Didier Marquet dans le cadre d'une collaboration entre MIRIAD/LIP6 et France Telecom R & D [Sempe *et al.*, 2002].

## 1.3 Le projet *MICRobES*

Ce travail de recherche s'inscrit dans le cadre du projet *MICRobES* [Drogoul and Picault, 1999] mis en place par Alexis Drogoul et les membres de l'équipe MIRIAD au laboratoire LIP6. Ce projet est consacré à l'étude des pré-requis nécessaires pour l'intégration de robots dans des environnements sociaux. En particulier, l'un des volets de ce projet porte sur les moyens donnés aux utilisateurs humains pour adapter, utiliser les robots et interagir avec eux. C'est dans ce volet du projet que s'inscrit mon travail de thèse.

## 1.4 Organisation de ce document

**Partie I - problématique détaillée et travaux existants :** La première partie de ce document est consacrée à l'analyse des difficultés rencontrées lors de l'acquisition de comportements situés dans les environnements réels. Les difficultés propres aux perceptions sont d'abord traitées dans le chapitre 2, celles relatives à l'environnement dans le chapitre 3 et celles concernant l'acquisition proprement dite sont abordées dans le chapitre 4. Vient ensuite, au chapitre 5 un récapitulatif des propriétés requises pour un système permettant l'acquisition de comportements, c'est également dans ce chapitre qu'est formulée *l'hypothèse générale* formant le cadre de cette thèse. Cette partie se termine par un examen des approches et travaux existants en relation avec l'hypothèse générale.

**Partie II - un modèle pour l'acquisition de comportements :** Le modèle d'acquisition que je propose (**MPL**) est décrit dans la deuxième partie. Celle-ci présente d'abord les hypothèses structurant le modèle et son cadre dans les chapitres 7 et 8. Le modèle proprement dit est décrit dans les chapitres 9, et deux variantes sont présentées dans les chapitres 10 et 11. Les compléments au modèle permettant l'utilisation des comportements sont donnés dans le chapitre 12.

**Partie III - Les expérimentations :** La troisième partie détaille les expérimentations effectuées. La plateforme robotique utilisée et la mise en oeuvre du modèle **MPL** sont décrites dans le chapitre 14. Les moyens employés pour structurer le champ perceptuel sont exposés

dans le chapitre 13. Finalement les expérimentations sont détaillées dans le chapitre 15 et discutées dans le chapitre 16.

**Partie IV - L'apprentissage interactif et ses perspectives :** La quatrième et dernière partie traite des perspectives, en particulier dans le chapitre 17 au travers d'une étude de l'apprentissage interactif reposant sur le modèle **MPL**. Les voies de recherches envisagées suivies d'une conclusion générale terminent ce document.

## Première partie

# Les difficultés de l'acquisition de comportements en environnement réel



## Avant-propos

Dans l'introduction j'ai indiqué mon principal objectif : l'acquisition de comportements d'usage de l'environnement pour un robot autonome. Pour illustrer le type de comportement visé on repensera à l'exemple de l'acquisition d'un comportement d'accostage (docking) d'un robot à une station de recharge. Dans cette partie je présente les principales difficultés que l'on rencontre lorsque l'on veut résoudre ce problème. Elles sont essentiellement dûes au fait que l'environnement du robot est ici le monde réel et non le monde clos et contrôlé d'un simulateur ou de l'environnement "artificiel" de l'ordinateur ou du modèle abstrait. Ce pas est important, car il rend difficile, voire impossible, de transposer les méthodes existantes, développés, pour la plupart, dans des univers uniquement logiciels.

Cette partie présente de façon détaillée les difficultés auxquelles j'ai été amené à faire face. Celles ci sont de trois types :

1. **les difficultés propres aux perceptions**
2. **les difficultés propres à l'environnement**
3. **les difficultés propres au processus d'acquisition**

L'examen détaillé de ces difficultés permet de circonscrire l'ensemble des systèmes envisageables pour l'acquisition de comportements et de formuler à la fin de cette partie l'hypothèse générale qui constituera le cadre des parties suivantes.



## Chapitre 2

# Les difficultés propres aux perceptions

### 2.1 Introduction

De prime abord, un robot est avant tout un ordinateur doté en entrée de plusieurs *senseurs* et en sortie *d'effecteurs* qui lui permettent, pour les premiers de percevoir son environnement et pour les seconds d'y agir. Cette conception classique du robot vu comme étant un ordinateur étendu pousse à concevoir des modèles dans lesquels perceptions et actions ne sont que des variantes plus sophistiquées des entrée-sortie traditionnelles de nos composants logiciels. Ainsi la perception est-elle considéré traditionnellement comme un sous-problème, certes difficile, mais qui sera réglé séparément par les disciplines du traitement du signal, de la vision ou encore celles concernant l'élaboration des capteurs. Il est traditionnel de penser que pour résoudre le 'vrai' problème, il vaut mieux l'aborder par 'en haut' en se concentrant sur l'élaboration des mécanismes cognitifs fondamentaux, sur les représentations abstraites propices à la résolution de problèmes ou sur l'étude de mécanismes d'apprentissage génériques. Les perceptions seront ensuite adaptées selon les besoins.

Pourtant, compte tenu de sa complexité intrinsèque, l'environnement perçu par le robot ne se laisse pas facilement résumer à un monde clairement étiqueté par des symboles propositionnels ou valués.<sup>5</sup> A tel point qu'il faut peut-être envisager d'aborder le problème par "en

---

<sup>5</sup>Et on pourrait reprendre pour la robotique une remarque plus générale faite par Jean Pierre Changeux



FIG. 2.1: Illustration classique de R. C. James extraite de l'ouvrage 'Vision' de D. Marr

bas", c'est à dire à partir des perceptions possibles [Brooks, 1991a]. Rien dans les perceptions du robot ne se traduit en termes explicites indiquant par exemple que 'ceci est une porte ouverte', que 'voici le couloir principal' ou que 'la station de recharge se trouve à gauche'. Ce problème que nous rencontrons dans le cadre de l'acquisition de comportements est une instance du problème général désigné par S. Harnad sous le terme de Symbol Grounding Problem. Cet auteur pointe la difficulté qu'il y a à construire une représentation symbolique dont les éléments soient reliés aux phénomènes du monde réel. Ainsi comme l'indique Harnad dans [Harnad, 1990] :

*"The standard reply of the symbolist (e.g., Fodor 1980, 1985) is that the meaning of the symbols comes from connecting the symbol system to the world "in the right way." But it seems apparent that the problem of connecting up with the world in the right way is virtually coextensive with the problem of cognition itself. (...) . Many symbolists believe that cognition, being symbol-manipulation, is an autonomous functional module that need only be hooked up to peripheral devices in order to "see" the world of objects to which its symbols refer (or, rather, to which they can be systematically interpreted as referring). Unfortunately, this radically underestimates the difficulty of picking out the objects, events and states of affairs in the world that symbols refer to, i.e., it trivializes the symbol grounding problem".*

---

[Changeux and Ricoeur, 2000] : nous vivons dans un univers "non étiqueté", qui ne nous envoie pas de messages codés. J'ai déjà protesté avec vigueur contre cette conception, chère à beaucoup de mathématiciens, d'un monde platonicien, fourmillant de formes et d'idée préétablies, d'un illusoire ciel étoilé décoré de propositions vraies, de rythmes harmonieux ou de maximes de bonne conduite. En fait, nous projetons sur un monde sans destin ni significations très précisément "des buts d'actions et des exigences de signification". C'est avec notre cerveau que nous créons des catégories dans un monde qui n'en possède pas, sauf celles déjà créées par l'homme.

Les perceptions des robots proviennent de senseurs intrinsèquement limités et subissent de nombreuses altérations dans les environnements réels. Sans aller jusqu'à une situation extrême comme celle la figure 2.1 - cependant pas tant éloignée des conditions réelles- il est difficile d'associer des objets 'connus' aux perceptions disponibles et de maintenir cette relation sur une longue période de temps. Plus encore que de percevoir correctement, la question qui se pose pour un concepteur de comportements est de savoir ce qu'il faut percevoir en vue de l'action à réaliser.

## 2.2 Les sens des robots

Avant tout, les robots perçoivent le monde au travers de leur senseurs. Dans la suite du texte on désignera par *senseur* les dispositifs matériels transformant les informations analogiques en provenance de l'environnement en informations numériques utilisables par l'ordinateur. On appellera de manière générale *perceptions* les interprétations des informations provenant des senseurs immergés dans l'environnement et on emploiera le terme *percepts* pour désigner les éléments de perception opérationnels extraits du flot sensoriel, c'est à dire utilisables par un logiciel.

Sans faire une revue exhaustive des senseurs disponibles, on peut lister leurs propriétés caractéristiques indépendamment d'un modèle particulier de capteur. On trouvera une revue détaillée des senseurs actuels dans l'ouvrage de G.Dudek et M. Jenkin [Dudek and Jenkin, 1999]. Bien qu'étant très divers, les senseurs peuvent être répartis en six grandes classes en fonction de la nature des informations qu'ils fournissent :

1. Senseurs de type présence.
2. Senseurs de type intensité/direction.
3. Senseurs de type distance/direction
4. Senseurs de type visuel
5. Senseurs de type environnementaux
6. Senseurs de type internes.

- **Les senseurs du type présence** sont les plus simples. Ils renseignent sur la présence ou l'absence d'un phénomène et donnent le plus souvent une information binaire indiquant le contact direct avec un obstacle ou la présence d'un dispositif dédié spécialement équipé - une station de recharge dotée d'un émetteur par exemple.
- **Les senseurs de la classe intensité/direction** informent sur l'intensité d'un phénomène dans une direction précise. Il peut s'agir par exemple de l'intensité lumineuse dans une direction, de l'intensité d'une odeur ou d'une source radio. De ce fait, ils sont adaptés aux stratégies de remontée de gradient où un robot se rapproche de la source d'une émission ( lumière, odeur) en choisissant toujours la direction d'intensité maximale.
- **Les capteurs de la classe distance/direction** renseignent sur la distance d'un phénomène dans une direction précise. On trouve dans cette classe les sonars, les capteurs infra-rouge et les télémètres laser. Les deux premiers donnent une information très imprécise sur la distance au plus proche obstacle dans une direction, le dernier donne cette même information avec une meilleure précision.
- **Les senseurs visuels** fournissent une image correspondant à la projection de la lumière provenant d'une zone délimitée de l'environnement sur un plan ou une ligne photosensible. L'image est constituée de points discrets (pixels) associant une information de couleur à une direction. De ce fait les senseurs visuels sont assimilables à l'assemblage d'un grand nombre de senseurs intensité/direction - au cours de l'évolution animale, il est admis que les senseurs visuels biologiques sont apparus suite à l'agrégation de plusieurs senseurs intensité/direction. Les capteurs de cette classe fournissent en général de  $10^3$  à  $10^7$  fois plus d'informations élémentaires que les senseurs des autres classes. Parmi eux on trouve les cameras numériques ou webcams usuelles éventuellement équipées d'objectifs spéciaux ou doublées pour la vision stéréoscopique.
- **Les senseurs environnementaux** fournissent une information globale telle que la température ambiante ou le degré d'hygrométrie.
- **Enfin les senseurs internes** fournissent une information de nature proprioceptive

<i>Classe</i>	<i>Exemples</i>
Contact	bumper, source infrarouge
Intensité/Direction	capteur d'odeur, photomètre
Distance/Direction	senseur infra-rouge, sonar , télémètre laser.
Visuel	camera mono-ou stéréo- scopique, webcam appareil photo numérique, caméra linéaire (avec une seule ligne de pixels).
Environnemental	thermomètre.
Interne	inclinomètre, gyroscope, odomètre, accéléromètre

TAB. 2.1: Récapitulatif des principales classes de senseurs.

telles que l'inclinaison, l'accélération, la distance parcourue. On trouve dans cette dernière classe les inclinomètres, les gyroscopes, les accéléromètres et les odomètres.

On imagine qu'une application robotique complète nécessite de mettre en jeu toute une panoplie de senseurs. Il devient difficile alors d'étudier et de comparer les systèmes compte tenu de l'étendue du bestiaire robotique et de la multiplicité des cas d'espèce. <sup>6</sup> Aucune de ces classes de senseurs ne facilite radicalement l'acquisition de comportement. Les problèmes rencontrés en robotique seront-ils réglés dès qu'apparaîtront des senseurs appropriés, plus performants, plus sensibles ou dotés d'une meilleure résolution? C'est peu probable, car si des progrès sont certainement réalisés, il ne supprimeront pas les imperfections intrinsèques des senseurs ni les altérations que subissent les perceptions dès lors qu'un robot est confronté à un environnement réel. De plus, comme le fait remarquer T. Smithers [Smithers, 1994], le fait de disposer de capteurs plus précis et plus riches, peut compliquer la tâche plutôt que la faciliter : un capteur plus riche permet de percevoir un monde plus riche, mais par là même

<sup>6</sup>Actuellement les travaux de recherche tendent à ce concentrer sur l'utilisation d'une seule classe de senseur et peuvent être répartis en trois groupes principaux. Une large partie des travaux s'effectue à l'aide de senseurs de type distance/direction dont les robots sont le plus souvent équipés en standard. Cette sur-représentation des senseurs distance/direction est déterminée par des contraintes économiques et technologiques. Ces senseurs sont avant tout peu onéreux et relativement simples à appréhender puisque les mesures qu'ils produisent sont constituées de deux ou trois valeurs réelles (valeur en mm et un ou deux angles). Leur utilisation a donné lieu à de nombreuses représentations des obstacles dans un espace à deux dimensions et la problématique qui a occupé depuis une vingtaine d'années beaucoup de chercheurs a été la résolution du problème de la navigation et la localisation d'un robot sur la base de carte et plan pré-établis [Yamauchi and Langley, 1997; Thrun *et al.*, 1997; Klingspor *et al.*, 1996]. Les senseurs visuels (caméras numériques, webcams, ..) sont cependant utilisés dans une par grandissante des travaux et ce encore pour des raisons technologiques et économiques. Enfin dans les approches de type Animat où l'inspiration biologique prime, beaucoup de travaux s'attachent à utiliser des senseurs particulièrement simples de type intensité/direction à l'instar des véhicules de V. Braitenberg [Braitenberg, 1984]. Ces senseurs trouvent leur correspondance dans le règne animal; en particulier chez les insectes.

plus complexe et qui nécessite des aptitudes de catégorisation plus élaborées.

### Senseurs imprécis, incertains et incomplets

Tous les senseurs existants, et ceux des robots n'y échappent pas, produisent des informations qui sont à la fois *imprécises, incertaines et incomplètes*.

Les informations fournies sont **incertaines** car les senseurs sont tous affectés par du *bruit*. Cette notion désigne tout ce qui n'est pas le signal désiré (et présuppose donc qu'un tel signal 'pur' existe). Le type de bruit auquel il est fait le plus souvent référence est le bruit gaussien car il est le plus facile à modéliser. Dans le cas du bruit gaussien, la mesure du signal pur est perturbée par des mesures bruitées réparties selon une loi normale. Cependant, à l'échelle spatiale et temporelle du robot, le bruit n'est pas uniquement gaussien. Il peut être non-stationnaire s'il varie avec le temps (on pensera à des personnes passant dans le champ visuel du robot), il peut être anisotropique si le champ perceptuel n'est pas affecté partout de la même manière (on songera à un éclat de lumière affectant un capteur visuel, ou un écho intempestif affectant un capteur sonar). Il peut être multimodal s'il se traduit par la combinaison de plusieurs gaussiennes ou encore non gaussien si sa distribution ne peut être approximée par une loi normale. Ces diverses formes de bruit sont rencontrés fréquemment dans l'environnement d'un robot réel.

Les informations fournies par les senseurs sont **incomplètes**. Le champ des senseurs est toujours limité, à une zone de l'espace ou à un plan. Les senseurs ne renseignent que sur quelques propriétés de l'environnement, la distance, la couleur ou encore la température. Ils ne fournissent qu'une vue partielle de l'environnement.

Les informations fournies sont **imprécises** car leur résolution est finie et leur calibrage ne peut être parfaite. Deux capteurs identiques pourront donner des valeurs différentes dans les mêmes circonstances. Ainsi une caméra sera plus sensible au rouge qu'une autre. Les capteurs distance/direction ont souvent une distance minimale en dessous de laquelle les mesures ne sont pas significatives.

## 2.3 Altération des perceptions

Le robot forme ses perceptions à partir de senseurs immergés dans un environnement donné. On peut penser pouvoir *reconnaître* des formes, des objets, des traits précis dans ce flot d'information mais la tâche est particulièrement difficile dans des robots mobiles. Pour un robot manipulateur fixe dont l'environnement est clairement délimité, aménagé et dans lequel le robot n'est appelé à opérer que dans un nombre de cas bien définis, les situations rencontrées sont peu nombreuses et peuvent être caractérisées avec précision. Il faut bien sûr prendre en compte le bruit et certaines déformations [Belaid and Belaid, 1992], mais il reste possible de reconnaître les formes qui se présentent par ailleurs dans le champ visuel du robot en nombre limité.

### Principales altérations

Dès lors que le robot se déplace dans son environnement, les perceptions du robot peuvent être dégradées de multiples manières. Ainsi un robot mobile est confronté à des altérations telles que :

- les déformations
- les disparitions
- les occlusions
- le bruit
- les ambiguïtés
- les enchevêtrements

Au travers d'un exemple simplifié dans lequel un robot mobile doit percevoir un cube posé au sol on peut se figurer les altérations qui surviennent systématiquement pour les deux classes principales de senseurs que sont les senseurs de type visuels et ceux de type distance/direction. Ces altérations rendent particulièrement complexe le processus d'interprétation des données sensorielles.

### Déformations

L'homothétie (changement d'échelle), le déplacement, la distorsion sont les déformations classiques de la perception visuelle. Elles sont dûes aux effets de perspective et aux change-

ments de point de vue. Si l'homothétie due à la projection sur un plan est le fait des senseurs visuels, on retrouve les autres déformations pour les senseurs distance/directions comme on peut le voir dans les figures 2.2 et 2.3.

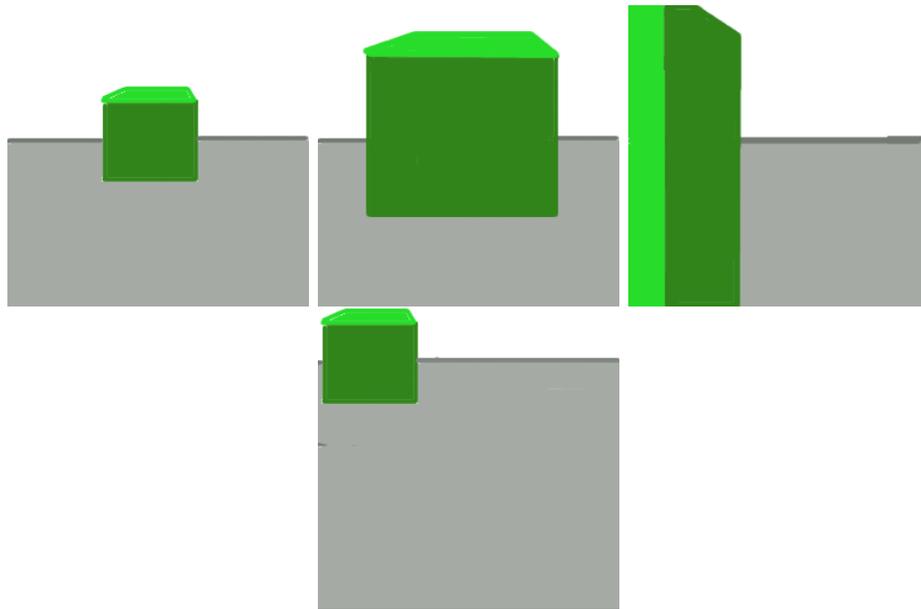


FIG. 2.2: *Déformations des perceptions visuelles*

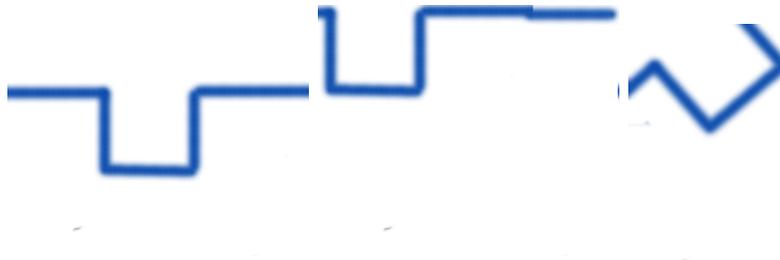


FIG. 2.3: *Déformation des perceptions distance/direction. Le même cube que celui de la figure 2.2 perçu par un senseur de type distance/direction dans différentes situations*

## Occlusions

Les objets perçus par un robot sont très souvent masqués, que ce soit totalement ou partiellement. Ce fait est d'ailleurs partagé par les robots et les humains : en regardant autour de soi, sur son bureau par exemple, on constate que très peu d'objets sont visibles entièrement. Ces occlusions peuvent se produire pour tous les types de senseurs comme le montrent les figures 2.4 et 2.5.

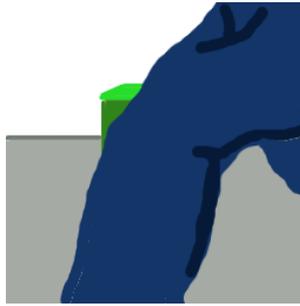


FIG. 2.4: Perception visuelle et occlusion. Ici un humain passe dans le champ visuel



FIG. 2.5: Perception distance/direction et occlusion dans une situation similaire à celle de la figure 2.4

## Disparitions

Comme dans la figure 2.6, un objet peut disparaître alors qu'il est présent. Il suffit par exemple qu'il occupe tout le champ visuel. Cette situation peut paraître très particulière mais elle se produit souvent et parfois systématiquement dans le cas d'une approche par exemple. On retrouve cet effet sur tous les senseurs, soit en raison de l'existence de distances minimales de perception, soit par effet de saturation.



FIG. 2.6: Disparition des perception visuelles

## Bruit

Les différents types de bruit déjà évoqués dans le paragraphe 2.2 affectent les perceptions de multiples manières. Le bruit de type gaussien comme celui des figures 2.7 et 2.9 est souvent généré par le capteur lui-même et dépend de la technologie utilisée. Les exemples de bruits non-gaussien comme ceux de la figure 2.8 sont en particuliers dus aux conditions d'éclairage pour les capteurs visuels. Il est d'ailleurs significatif que les organisateurs de la RoboCup 2002 tentent de se prémunir de ce type de problème par voie réglementaire<sup>7</sup>.

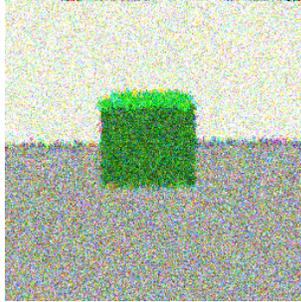


FIG. 2.7: Perception visuelle bruitée - bruit gaussien

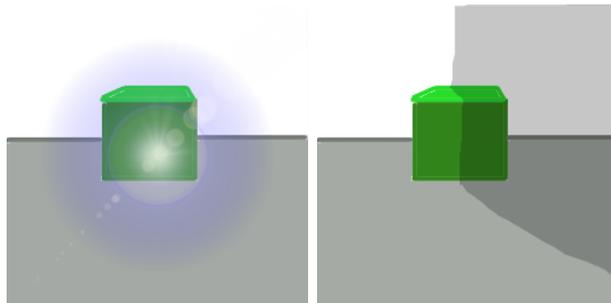


FIG. 2.8: Perception visuelle bruitée - bruit non gaussien

## Ambiguïté (aliasing)

Des situations différentes peuvent conduire à des perceptions similaires et par là ambiguës. Comme le montre la figure 2.10, une caméra noir et blanc renvoie la même image (rectangle au centre de l'image) pour des contextes très différents.

---

<sup>7</sup>Extrait du règlement : "Camera lights are prohibited. Light strobes and other devices that add illumination are prohibited during photography. (1,000 lux brightness is available on all courts.) ... Refrain from wearing brightly colored clothing, particularly primary colors : red, blue, yellow, orange. These colors are affected to the robots for their visual detection."

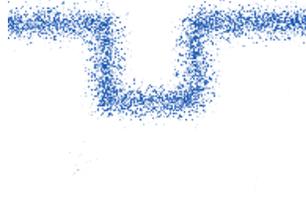


FIG. 2.9: Perception distance/direction et bruit gaussien

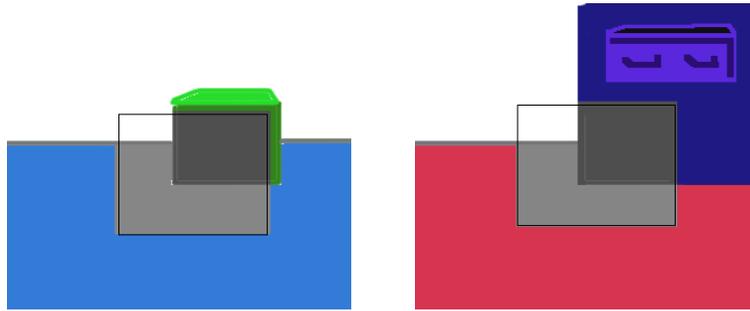


FIG. 2.10: Ambiguïtés des perception visuelles

### Enchevêtrement (cluttering)

Les éléments constituant le monde du robot ne se présentent pas les uns bien séparés des autres comme dans une base de donnée d'images mais le plus souvent enchevêtrés et mêlés les uns aux autres. Dans ce cas les différentes altérations telles qu'occlusion, ambiguïté et bruit sont multipliées et la complexité de la tâche consistant à percevoir des éléments connus mais enchevêtrés avec d'autres s'accroît d'autant.

#### 2.3.1 Exemples réels

Les images de la figures 2.11 donnent une idée de ce que perçoit un robot dans des situations réelles. Elles ont été obtenues avec à partir de la caméra monoscopique couleur d'un robot mobile. La caméra pourrait être d'encore meilleure qualité, le champ visuel plus large ou la résolution plus précise, mais cela ne changerait pas fondamentalement la réalité des altérations rencontrées.

La figure 2.12 montre plusieurs images captées lors de la progression du robot vers une porte. On y retrouve les altérations telles que les déformations, les occlusions, l'ambiguïté entre les différentes portes, la disparition de la porte lorsqu'on la passe et le bruit provenant des reflets et des ombres.

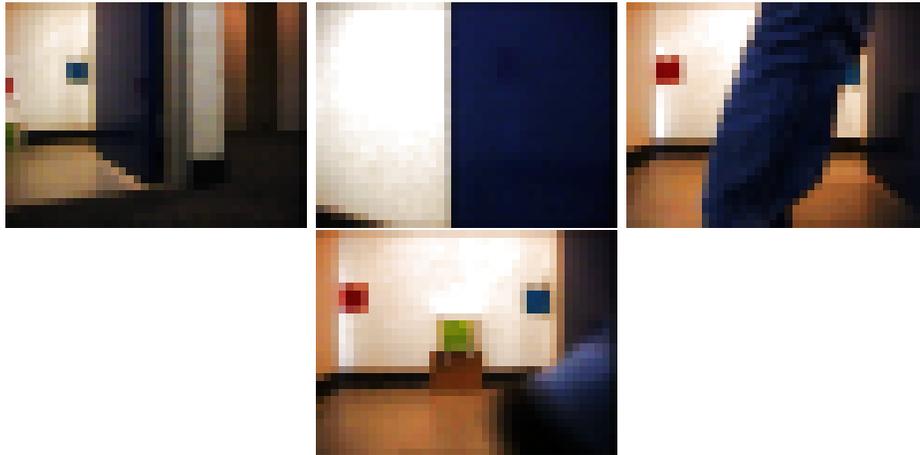


FIG. 2.11: Images provenant de la camera d'un robot réel.



FIG. 2.12: Approche d'une porte - images provenant de la caméra d'un robot réel.

Les perceptions distance/direction posent des problèmes similaires. Au cours d'expériences préliminaires, j'ai pu en constater les effets et vérifier que les approches statistiques améliorent les perceptions sans pouvoir pour autant rendre triviale la reconnaissance ou l'identification des objets. La figure 2.13 montre les mesures sonar récoltées après qu'un robot mobile se soit déplacé sur une douzaine de mètres dans un couloir large de 3 mètres. Les mesures ont été reportées dans un repère à deux dimensions en tenant compte des déplacements du robot. L'information provenant des capteurs peut être certainement améliorée par des dispositifs permettant de fusionner les mesures et d'intégrer plusieurs mesures successives comme le font par exemple les grilles d'occupation. Cette technique [Moravec, Summer 1988; Elfes, 1989] est très utilisée pour obtenir une représentation en deux dimensions de l'espace autour du robot [Thrun *et al.*, 1997; Yamauchi and Langley, 1997] et sert de support pour la cartographie et la navigation de nombreux systèmes robotiques autonomes. Une grille d'occupation représente la probabilité de présence d'obstacles dans une grille discrète. Elle est obtenue à partir de mesures provenant de capteurs de distance (sonars ou télémètre laser), et d'un modèle probabiliste de

ces capteurs. La formule de Bayes est utilisée pour l'intégration de nouvelles mesures. De tels mécanismes montrent qu'il est possible d'améliorer les mesures brutes mais néanmoins le bruit demeure, de nombreux détails sont absents et des obstacles fantômes apparaissent causés par la conformation de l'environnement (généralisant des échos intempestifs).

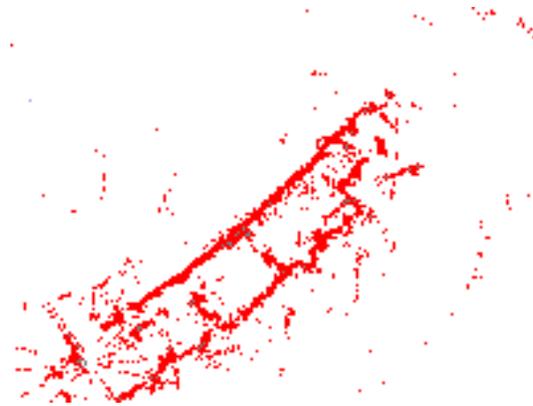


FIG. 2.13: Mesures sonar réelles obtenues pendant le déplacement d'un robot sur une douzaine de mètres et reportées sur un plan.

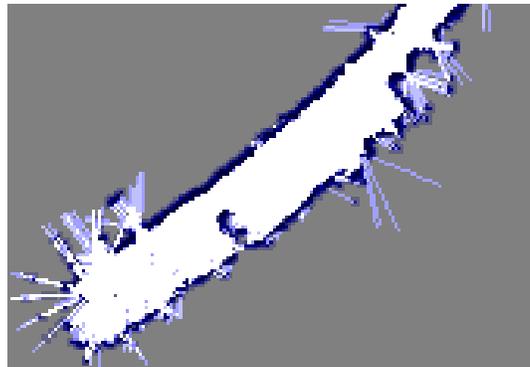


FIG. 2.14: Grille d'occupation obtenue à partir des mesures sonar réelles de la figure précédente.

## 2.4 Difficulté d'associer objet et perception

Ce que nous percevons immédiatement comme un objet ayant ses limites, sa cohérence interne, son identité parfaitement dissociée des autres éléments de l'environnement apparaît au robot sous de multiples facettes difficiles à intégrer en une seule entité; il en est ainsi du cube des exemples précédents mais aussi de la plupart des objets souvent plus complexes qu'un robot peut rencontrer. Le statut d'objet n'est d'ailleurs pas évident, comme le fait remarquer David Marr dont les travaux ont fondé pour une large part la discipline de la vision [Marr, 1982] : *"Est-ce qu'un nez est un objet ? et une tête ? Est ce toujours un objet si elle est rattachée à un corps ? Et un homme à cheval ? Ces questions montrent les difficultés qu'on rencontre quand on essaie de formuler ce qui devrait être reconnu comme une région dans une image"* .

S. Coradeschi et A. Saffioti dans [Coradeschi and Saffioti, 2001] définissent le problème de l'association des données sensorielles à des éléments connus comme étant celui de *l'ancrage*, c'est à dire selon eux : *"Le processus par lequel est crée puis maintenu la correspondance entre les symboles et les percepts relatifs à de mêmes objets physiques"*. Cette définition pragmatique à ses limites puisqu'elle présuppose que les symboles peuvent être constitués a-priori indépendamment des percepts. Par ailleurs elle sous-entend comme nécessaire que le monde soit perçu par les systèmes artificiels comme composé d'un ensemble d'objets <sup>8</sup>.

Les moyens pour la reconnaissance, la catégorisation et l'identification d'objets sont nombreux et certaines méthodes donnent d'excellent résultats d'autant plus que les problèmes sont

---

<sup>8</sup>Une critique radicale de cette vision de choses est formulée par exemple par J.V Stone [Stone, 1993] : *Linguistic description of the physical world tend to be expressed in terms of objects an their associated properties and processes. An example is : 'The green stone is sinking'. Here the primary descriptor is the stone with green (property) and sinking (process) being 'attached' to the stone. However this lingusitic description belies a bias which may bear little relation to the type of quantities computed by a visual system. Within the visual system the motion, colour and spatio-temporal integrity of the stone are not necessarily subsidiary to each other, they are simply computable attributes of a physical scenario. I believe that this linguistic bias has influenced much work in computer vision and has retarded progress by identifying computer vision with object identification, rather than with what vision is used for in biological vision systems. More recently , the apparent intractability of computer vision as object identification, and the consequent lack of practical use of much of computer vision work, has led to a re-appraisal of what vision is for. This move away from conventional computer vision and toward animated vision is motivated by the modest practical success of computer vision systems. It is now accepted by some that computer vision was asking the wrong questions. Rather than asking 'How can we get a machine to name objects ?' perhaps it should have been asking 'What is vision for ?'* .

précis et traités dans un cadre bien défini - dans le cadre de l'imagerie médicale ou satellitaire par exemple, ou encore dans celui des chaînes de production industrielles. Ces moyens peuvent être répartis en trois principales catégories :

- Ceux visant à la reconnaissance d'un modèle à partir des perceptions.
- Ceux visant à la classification supervisée des perceptions à partir de données empiriques.
- Ceux visant à la catégorisation automatique des perceptions.

La reconnaissance à partir d'un modèle constitue l'approche la plus classique, le modèle étant décrit sous forme géométrique en 2 ou 3 dimensions. On trouvera une revue des méthodes classiques dans les ouvrages suivants [Sonka *et al.*, 1993] pour l'aspect vision et [Belaid and Belaid, 1992] et [Duda and Hart, 1973] pour la reconnaissance proprement dite. On notera que les techniques sous-jacentes aux processus de reconnaissance sont souvent la segmentation et la détection de contour qui sont connues pour être sensibles aux paramétrages et conditions d'exploitation. On trouve des applications de ces techniques pour la reconnaissance de visage ou la surveillance du trafic routier à partir de cameras vidéo fixes [Worrall and Baker, 1993]. Des travaux récents [Isard and Blake, 1996] ont par exemple montré qu'il est possible de pister (tracking) un objet mobile avec précision dans une séquence vidéo complexe et ce sur la base d'un modèle graphique, après avoir effectué un apprentissage statistique d'un modèle de mouvement de l'objet. Il s'agit cependant de pister quelque objets parmi tout ceux présents dans la scène, alors que le robot doit tenir compte de tout les objets se trouvant autour de lui ne serait-ce que pour les éviter.

La reconnaissance d'objets ou de scènes peut aussi être vue comme un problème de classification supervisée à partir d'un ensemble d'images, chaque image de l'ensemble d'apprentissage étant étiquetée par un intervenant humain au moyen d'un label indiquant la classe désirée de l'image. Les techniques d'apprentissage numérique utilisables dans ce cas sont par exemple les fonctions à base radiale [Poggio and Edelman, 1990], les perceptrons multi-couches [Bishop, 1995], ou plus récemment les support vector machines (SVM) [Chapelle *et al.*, 2001; Osuna *et al.*, 1997]. On trouvera une description technique détaillée de ces méthodes dans [Haykin, 1999]. Ces méthodes requièrent un nombre d'exemples important et un temps d'apprentissage relativement long si on l'envisage dans le cadre de l'interaction avec un robot. En

entrée des algorithmes de classification on privilégie des traits (features) extraits des images qui soient autant que possible invariants aux transformations courantes telles que les translations, rotations et homothétie. Il est particulièrement notable que des traits extrêmement simples comme les histogrammes de couleurs [Swain and Ballard, 1991] ou des 'Blob' de couleur [Carson *et al.*, 1997] aient rencontré beaucoup de succès en indexation d'image.

Les enjeux économiques actuels font que l'intérêt se porte avant tout sur les méthodes permettant de faciliter l'accès aux grandes bases de données d'images (Image Retrieval by Content, Image Indexation). Cependant *la nature des images utilisées dans les tâches d'indexation et de classification est très différente de celles effectivement rencontrées par un robot*. Ce dernier se déplace de façon continue dans son environnement, il perçoit les objets depuis des **postures** qui sont proches les unes des autres (lors de l'approche d'une station de recharge, celle-ci se présentera successivement sous des angles très différents). Le robot doit donc décider d'actions différentes en fonction de perceptions relativement similaires. De ce fait, les contraintes imposées aux algorithmes de classification d'images dans le cadre de l'indexation, par exemple, ne peuvent être reprises pour la classification des perceptions d'un robot. Dans le cas du robot, il faut distinguer des perceptions similaires et en déduire des actions différentes, alors que dans le cas de l'indexation, il faut au contraire agréger dans une même classe des images similaires.

Plutôt que de reconnaître un objet ou une scène sur la base d'informations données par un humain on peut au contraire chercher à catégoriser automatiquement les perceptions du robot dans un cadre non-supervisé. Cette approche est la plus proche de théories actuelles en neurophysiologie et psychologie [Riesenhuber and Poggio, 2000][Goldstone, 1998] [Edelman and Duvdevani-Bar, 1997] [Linsker, 1988]. Elle repose sur un mécanisme d'auto-organisation effectuant une différenciation progressive des percepts. Comme l'indique R.L. Goldstone "*A major mechanism of perceptual learning is for percepts to become increasingly different from each other*". Ou encore la maximisation de l'information mutuelle entre perception et action <sup>9</sup>. Dans le cadre de la robotique les cartes de Kohonen [Kohonen, 1997] utilisant un mécanisme de différenciation ont par exemple été utilisées pour catégoriser automatiquement les perceptions provenant de capteur distance/direction en vue de la cartographie [Nehmzow and Smithers,

---

<sup>9</sup>L'information mutuelle est une notion provenant de la théorie de l'information et correspond à une mesure de la réduction de l'entropie d'une variable aléatoire connaissant une autre variable aléatoire.

1991; Kurz, 1996] ou de la communication inter-robots [Hugues, 2000b]. Cependant, il existe une infinité de manière de catégoriser automatiquement des perceptions et le problème se pose alors de trouver un critère qui soit pertinent pour la tâche considérée..

### 2.4.1 Que faut-il percevoir ?

Le robot humanoïde présenté par D.C. Bentivegna et C.G Atkeson [Bentivegna and Atkeson, 2000] joue au hockey de table debout en face d'un joueur humain. Le jeu n'est possible qu'après une phase d'apprentissage. En voyant son aisance, on s'attend à le voir se lever et pourquoi pas aller jouer au billard ou à la roulette sur une table voisine. Il s'agit d'une prouesse qui semble bien au-delà de l'acquisition d'un comportement d'usage de l'environnement. Cependant les concepteurs ont tout de même largement facilité le travail du robot en le dotant de perceptions parfaitement adaptées au problème : ce robot perçoit en effet le monde via une caméra vidéo située à la verticale de la table de hockey. De cette caméra est obtenue une image dont sont extraits 5 éléments comportant des couleurs facilement identifiables : les quatre coins de la table et le palet. La perception du robot est donc extrêmement contrainte, n'importe quelle situation pouvant être résumée par la position et le vecteur vitesse du palet.

Dans le cas décrit ci-dessus, comme dans de nombreux travaux en robotique, **le support perceptuel du comportement**, c'est -à-dire l'ensemble des traits en fonction desquels les actions sont déclenchées, est réduit au minimum pour correspondre à la tâche considérée. Il est clair que, dans le cas d'un robot autonome mobile, cette réduction n'est pas toujours envisageable. S'il faut qu'un expert examine l'environnement, le comportement envisagé et les capacités du robot et en déduise ce qu'il faut percevoir en vue de l'action à effectuer, on ne peut plus véritablement parler d'acquisition de comportement. Ainsi, plus encore que de percevoir correctement des éléments connus, *le problème essentiel est de savoir ce qu'il faut percevoir pour mener à bien une action particulière.*



## Chapitre 3

# Les difficultés propres à l'environnement

### 3.1 Introduction

Reprenons l'exemple du comportement d'accostage déjà cité, en le considérant comme un prototype de comportement utile dans des environnements réels. Son étude peut être envisagée dans différents cadres : (1) celui d'un monde 2D simulé dans un ordinateur et où auront été modélisés les obstacles, la station, les senseurs; (2) avec un robot réel dans une arène fermée aux conditions environnementales bien contrôlées - luminosité, couleurs, perceptibilité des obstacles, sol sans aspérités etc.; (3) dans un environnement non préparé réellement fréquenté par des humains et conçu pour eux tels que des locaux d'entreprise ou de laboratoire; (4) à la station de métro Auber aux alentours de 18h près des guichets.

Dans son acception la plus générale le terme 'environnement' correspond à l'ensemble des conditions extérieures naturelles et culturelles susceptibles d'agir sur un robot. Les conditions rencontrées par les robots autonomes sont à long terme celles de leur environnement de destination tels que les locaux de bureaux, les ateliers, les zones de stockage, les établissements publics tels que les hôpitaux ou encore les musées, les domiciles privés, etc.. au moins pour ce qui est des environnements intérieurs. Cependant les environnements auxquels sont confrontés les robots autonomes d'aujourd'hui sont le plus souvent les locaux des laboratoires de robotique pour les robots réels ou les mondes en deux dimensions pour les robots virtuels décrits



FIG. 3.1: Les environnements d'expérimentation **Simulés** sont très fréquemment utilisés dans les expérimentations robotiques. Pour la plupart d'entre eux le réalisme est minimal (à gauche le simulateur de la RoboCup) de plus en plus ils intègrent la 3D (à droite Webbot de la société Cyberbotics.)

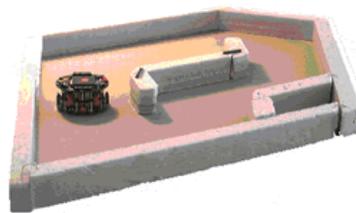


FIG. 3.2: Les environnements d'expérimentation de type **Arène** (ici celui de Floreano et Mondada à l'EPFL) sont préparés pour une expérience précise, ils présentent aux robots des situations très simplifiées.

dans un nombre important de travaux. Beaucoup plus rares sont les travaux qui ont donné lieu à des expérimentations significatives en environnements réels [Thrun *et al.*, 1999][Burgard *et al.*, 2000].

Intuitivement, une des grande différences entre un système robotique et les systèmes logiciels réside dans la nécessité de prendre en compte un environnement complexe réel. Pour pouvoir comparer les systèmes, choisir et valider des hypothèses, il faut d'abord pouvoir caractériser précisément leurs environnements de destination. A partir de ces principales caractéristiques il devient possible de vérifier la validité de chaque approche pour les problèmes considérés.

Des architectures génériques peuvent être conçues dans des environnements (simulés par exemple) qui ne soient pas les environnements définitifs. Mais, à moins de présupposer que toutes les caractéristiques d'un problème réel puissent être retrouvées dans un monde minimal abstrait, il faut indiquer quelles sont les étapes permettant aux solutions proposées



FIG. 3.3: Les environnements d'expérimentation de type **Réaliste** (ici l'un de ceux utilisés pour mes propres expérimentations) correspondent à des lieux peu ou pas préparés et réellement utilisés par des utilisateurs humains.

de fonctionner dans les environnements de destinations. Dire que ces étapes ne sont qu'affaire d'ingénierie ultérieure et d'adaptations ponctuelles reviendrait à dire que l'adaptation au monde réel est un problème trivial.

## 3.2 Caractérisation classique de l'environnement

La notion d'environnement en intelligence artificielle est apparue logiquement en même temps que celle d'agent. Dans leur ouvrage "Artificial Intelligence : A Modern Approach", S. Russel et P. Norvig [Russel and Norvig, 1995] fournissent une typologie des propriétés caractérisant l'environnement d'un agent.

### Types d'environnement d'un agent autonome (d'après S. Russel et P. Norvig)

*Environnement inaccessible/accessible* - l'environnement d'un agent lui est accessible s'il a accès à une description complète de l'état de l'environnement. Dans ce cas, il n'a pas à maintenir d'état interne et dispose à tout moment des informations suffisantes pour choisir ses actions. Dans le cas contraire l'environnement est inaccessible.

*Environnement discret / continu* - un environnement est discret si les perceptions et les actions de l'agent y sont clairement définies et en nombre limité. Il est continu dans le cas contraire.

*Environnement épisodique / non-épisodique* - L'environnement de l'agent est épisodique si celui peut y procéder par étape distinctes sans qu'il lui soit nécessaire de tenir compte des

étapes précédentes.

*Environnement statique / dynamique*- Si l'environnement peut changer pendant que l'agent décide de l'action à entreprendre, l'environnement est dynamique et il est statique dans le cas contraire.

*Environnement déterministe / non déterministe* - L'environnement de l'agent est déterministe si son état suivant peut être déterminé simplement à partir de l'état courant .

Qu'en est il pour un robot réalisant un comportement situé tel qu'un accostage dans un environnement réaliste ou réel ? Son environnement est d'abord *inaccessible* ou encore *partiellement observable* selon les termes consacrés en robotique [Kaebling *et al.*, 1995]. Du fait de sa perception limitée, le robot ne perçoit qu'une partie de son environnement. Compte tenu de l'ambiguïté perceptuelle, il ne peut pas toujours déterminer celle dont il s'agit. Rien ne ressemble plus à un mur blanc qu'un autre mur blanc.

L'environnement du robot est *continu*. S'il dispose d'une caméra ou d'un senseur de type télémètre laser, ses perceptions sont nombreuses et difficiles à distinguer les unes des autres. Ses actions se sont pas clairement discrétisées ni en nombre limité. Le robot réel ne saute pas de case en case comme beaucoup de robots simulés : étant doté de simplement deux roues motorisées, la gamme de ses actions possibles dans une période de 10 secondes est quasiment infinie.

Si le robot de la chaîne industrielle reste dans un environnement épisodique en traitant les pièces les unes après les autres, l'environnement du robot mobile est *non épisodique*. Chaque étape d'un comportement est dépendante des précédentes. Il devra par exemple s'être approché d'une porte avant de tenter de la passer.

L'environnement du robot est *dynamique* car des événements imprévisibles peuvent se produire alors qu'il réalise un comportement. Dans un environnement réaliste ou réel, un humain peut passer devant le robot alors qu'il s'approche de son objectif, un colis peut avoir été déposé sur son chemin ou les conditions de luminosité peuvent tout à coup changer.

Le robot réel est enfin confronté à des environnements *non déterministes*. Ses senseurs sont incertains et le fonctionnement de ses effecteurs est imprédictible. Il peut au mieux déterminer l'effet de ses actions de manière probabiliste s'il dispose d'un modèle du fonctionnement de ses effecteurs.

### 3.3 Caractéristiques supplémentaires de l'environnement

Les caractéristiques précédentes sont avant tout pensées pour la description de l'environnement d'un agent logiciel et elles ne suffisent pas pour décrire l'environnement d'un robot réel. Je propose de rajouter à cette liste de nouvelles caractéristiques mettant l'accent sur la spécificité de l'environnement réel. Ces caractéristiques ne sont pas des variantes ou des cas d'espèces des propriétés précédentes mais indiquent bien qu'il y a un changement de monde radical lorsque l'on passe de l'agent logiciel à l'agent robot. Les caractéristiques que je propose de rajouter sont les suivantes :

- Environnement étiquetable/ non étiquetable
- Environnement avec référentiel / sans référentiel
- Environnement partagé / réservé
- Environnement modifiable / figé
- Environnement dangereux / inoffensif.
- Environnement avec lois /sans lois

*Environnement étiquetable/non étiquetable* L'environnement est étiquetable si l'agent peut associer une étiquette symbolique aux objets qui l'entourent telles que 'porte' , 'chaise', 'couloir', etc. Comme on l'a vu, dans un environnement réaliste, il est difficile d'associer un symbole à un ensemble de perceptions. L'environnement réel peut comporter quelques étiquettes, si on a doté le robot des dispositifs *ad-hoc* de reconnaissance/classification mais ces dispositifs sont nécessairement limités et en nombre restreint. L'environnement du robot réel est donc au mieux partiellement étiquetable.

*Environnement avec référentiel / sans référentiel* - L'environnement peut comporter un référentiel spatial par rapport auquel sont exprimés la position des objets et du robot lui-même. Le robot aura accès à un référentiel si par exemple il dispose d'une carte et qu'il peut s'y localiser avec précision. Ce problème de la localisation est l'un des principaux sujets de recherche en robotique, partiellement résolu pour certains environnements et lorsque le robot

est doté de senseurs distances/direction précis (télémètres laser) [Thrun *et al.*, 2001]. Si les positions des objets sont exprimées par rapport à celle du robot le référentiel est dit *égocentrique*, si au contraire elle le sont par rapport à une référence extérieure au robot, le référentiel est dit *allocentrique*. Il semble que, dans le règne animal, seuls les primates aient un référentiel allocentrique [Berthoz, 1997].

*Environnement partagé / réservé* L'environnement du robot lui est réservé s'il ne le partage avec aucun robot ou humain. Le partage de l'environnement implique qu'il y a une compétition entre agents pour utiliser les ressources, en particulier les ressources spatiales telles que les passages et les accès [Fontan and Mataric, 1998][Simonin and Ferber, 2001]. La dynamicité de l'environnement est augmentée dans un environnement partagé, les situation d'occlusion des perceptions sont plus fréquentes et moins prévisibles. Enfin, un environnement partagé, il est nécessaire de tenir compte de la dangerosité de l'agent robot pour les autres agents.

*Environnement modifiable / figé* C'est une particularité des environnements réels, et humains particulièrement, de pouvoir être changés profondément et très rapidement - par exemple lorsque l'on réaménage une pièce, un atelier, ou lorsque des dispositifs nouveaux sont installés. Cette caractéristique est d'un autre ordre que la dynamicité, car elle suggère des modifications topologiques radicales et pose le problème du coût de la réadaptation du robot à son nouvel environnement. Un système qui nécessiterait pour cela une équipe de chercheurs ne pourrait convenir.

*Environnement dangereux / inoffensif.* Pour un agent logiciel, l'environnement est rarement considéré comme dangereux. Par contre l'environnement réel ou réaliste est véritablement dangereux pour le robot et les occasions de destruction du robot ou du mobilier sont multiples. Pour passer outre, il ne suffit pas de doter le robot d'un classique comportement d'évitement d'obstacle, car, s'il doit comme dans notre exemple accoster une station de recharge il lui faudra à un moment inhiber ce réflexe et prendre des risques. La dangerosité d'un environnement à pour conséquence très importante de rendre irréaliste les stratégies de type essai-erreur à grande échelle.

<i>Caractéristique</i>	<i>Illustration</i>
dynamique	survenue d'humains, d'autres robots.
non épisodique	les robot sont mobiles.
non déterministe	les consequences des actions des robot sont partiellement indéterminées
continu	les actions et perceptions des robots sont continues.
non accessible	les robot n'ont pas une représentation fiable et complète de leur environnement, ils sont situés.
non (ou partiellement) étiquetable	le coût d'etiquetage est trop important.
modifiable	apparition de nouveau objets, nouveau meubles, (aspirateur, colis).
avec référentiel partiel	eventuellement quelques balises pour se repérer.
partagé	lieu partagés par des humains et des robots
dangereux	présences d'escaliers, d'ascenseurs.
avec lois	ici on ne passe pas.

TAB. 3.1: Caractéristiques d'un environnement réel social.

*Environnement avec lois /sans lois* Dans un environnement sans loi, un robot n'est limité que par ce qu'il lui est possible physiquement de faire. Au contraire s'il coexiste avec des humains un robot devra éviter certains endroits et certaines actions et obéir à un ensemble de lois. Un environnement partagé et doté de lois peut être considéré comme étant un environnement *social* [Drogoul and Zucker, 2002].

L'environnement du projet *MICRobES* (cf. paragraphe 1.3) s'est avéré être un exemple type *d'environnement réel social*. Ses caractéristiques sont résumées dans le tableau 3.1.

### 3.4 Percevoir et agir dans l'environnement

Pour un robot, l'environnement n'est accessible qu'aux travers de ses perceptions et elles n'ont de sens qu'au vu des actions qu'elle lui permettent d'envisager. C'est avant tout l'association environnement/perception/action qui compte pour le robot. La perception n'est pas à étudier pour elle-même mais en vue de l'action qu'elle lui permet de réaliser.

L'éthologue Jacob Von Uexkull distinguait dans l'environnement d'un animal son *entourage*, c'est-à-dire les objets du monde considérés d'un point de vue objectif, et son *milieu*, formé de ce qu'il peut percevoir et dont il peut faire usage. Ainsi, pour un même entourage

des agents différents évoluent dans des milieux différents <sup>10</sup>. Pour les agents logiciels, une notion similaire, celle de *LifeWorld*, est proposée par Philip Agre et Ian Horswill : "*While there is an objective material environment, the agent does not directly deal with all of this environment's complexity. Instead it deals with a functional environment that is projected from the material environment. That projection is possible because of various conventions and invariants that are stably present in the environment or actively maintained by the agent. The lifeworld should be understood as this functional world together with the projection and the convention that create it*" [Agre and Horswill, 1997] .

Ainsi le problème est bien de déterminer la relation perception/action pour un robot spécifique dans un *contexte* spécifique : *Que peut percevoir ici ce robot ? Quelles actions lui sont possibles ici ? Quelle perception aide à réaliser quelle action ?*

Il faut de plus garder à l'idée que le temps du robot n'est pas celui de l'ordinateur. En effet le robot ne doit pas réagir le plus vite possible mais *réagir à temps*, avant par exemple de rencontrer un obstacle ou que la situation observée ne change. Cette caractéristique de *système réactif en temps réel* est d'autant plus importante que la dynamicité de l'environnement est forte. De ce fait, tous les systèmes reposant entièrement sur un schéma dans lesquels un robot doit d'abord se représenter entièrement son environnement à partir de ses perceptions pour ensuite déterminer la séquence d'actions à réaliser sont voués à échouer dans les environnements réels.

Les caractéristiques de l'environnement réel ont des conséquences directes sur les façons d'envisager et de formuler les actions du robot. Le déroulement des actions d'un robot en vue d'un but est considéré classiquement comme la réalisation d'un plan [Nilson, 1973]. Comme le rappelle R. Pfeifer la réalisation d'un plan s'apparente à la résolution d'un problème [Pfeifer, 1996] : "*The planning model in cognitive science treats a plan as a sequence of actions desi-*

---

<sup>10</sup>Extrait de [Uexkull, 1965] "*Examinons, par exemple, la tige d'une fleur sauvage et demandons-nous quel rôle elle joue dans les milieux suivants : 1) Dans le milieu d'une jeune fille qui cueille des fleurs et en fait un bouquet multicolore qu'elle épingle à son corsage ; 2) Dans le milieu de la fourmi qui utilise la texture unie de la surface de la tige comme un carrelage idéal pour aller vers sa nourriture, vers les pétales de la fleur ; 3) Dans le milieu de la larve de cigale qui perce le canal médullaire de la tige et l'utilise comme poste de pompage pour construire les murs fluide de sa maison aérienne ; 4) Dans le milieu d'une vache qui arrache feuilles et tiges pour les introduire dans son large mufle et s'en nourrir. . . . Chaque milieu constitue une unité fermée sur elle-même, dont chaque partie est déterminée par la signification qu'elle reçoit pour le sujet de ce milieu. Selon la signification pour l'animal, la scène où il joue son rôle vital englobe un espace plus ou moins grand, dont les lieux sont entièrement dépendants, en nombre et en grandeur, du pouvoir discriminant des organes sensoriels de cet animal*".

gned to accomplish some preconceived end. The model posits that action is a form of problem solving, where the actor's problem is to find a path from some initial state to a desired goal state, given certain condition along the way."

Ce point de vue a largement été critiqué depuis une décennie [Agre, 1988], [Suchman, 1987], [Maes, 1990]. Les principales critiques portent sur le fait que les environnements réels sont dynamiques et changeants et que par conséquent la réalisation d'un plan après une phase de délibération a beaucoup de chance de rencontrer, en temps réel, une situation non prévue. De plus la planification, est elle aussi confrontée au "symbol grounding problem" puisqu'elle manipule des symboles qui, d'une manière ou d'une autre, doivent être mis en relation avec les éléments réels.

A l'opposé de l'action faisant partie d'un plan on trouve la notion d'*action située*. Toujours selon R. Pfeifer, la définition du caractère situé d'un agent peut être décrit comme suit : "*Situatedness*" roughly means the following. First it implies that the world is viewed entirely from the perspective of the agent (not from the observer's perspective). Second a situated agent capitalizes on the system-environment interaction. Its behavior is largely based on the current situation rather than on detailed plans. It only focuses on relevant aspects of the situation. And third, a situated agent is not merely reactive, but brings its own experience to bear on the current situation. . . . Because of these properties, situated agents can act in real time."

Les systèmes robotiques doivent dans tous les cas intégrer une stratégie réactive dans leur architecture pour faire face à des situations imprévues lorsqu'elle se présentent. Les processus de décision, s'ils sont délibératifs, doivent au moins intégrer des mécanismes de type *any-time* [Dean and Boddy, 1988] permettant d'utiliser à tout moment les résultats intermédiaires des algorithmes de décisions. Par ailleurs, les capacités d'actions d'un système mobile sont limitées par la vitesse à laquelle sont prises en compte les perceptions (*visual throughput*). Plus un système est rapide plus il doit traiter les perceptions rapidement. *Une faible complexité en temps pour les algorithmes de décision n'est pas ici simplement souhaitée mais elle constitue une condition sine qua non de l'existence du système.*

Si la planification, dans sa version classique, rencontre des difficultés dans les environnements réels [Brooks, 1991b], elle ne peut cependant être complètement éliminée. Elle permet en effet de spécifier clairement le contrôle d'un système robotique à un haut niveau d'abstrac-

tion. Une fois passée la critique radicale sur la planification de la fin des années 80, [Suchman, 1987], [Agre, 1988], il est clair que le point de vue des systèmes situés ne peut à lui seul répondre au problème de la conception des systèmes robotiques et qu'il doit être associé d'une manière ou d'une autre à *une stratégie explicite de contrôle en vue de l'intégration dans une application complète*. Par ailleurs, sur un plan plus théorique, on peut considérer que les séquences d'actions complexes ont une organisation quasi-syntaxiques [Calvin, 1990] et qu'il est fondamental d'étudier les liens possibles entre le niveau de l'action située et celui de la séquence structurée d'actions.

En résumé, lors de la réalisation d'un comportement situé, l'action dépend du contexte dans lequel elle s'effectue ainsi que du robot qui l'effectue, et elle s'inscrit dans un temps réel pendant lequel des événements imprévus peuvent se produire. Elle nécessite la perception non pas de tout ce qui se présente mais des éléments qui aident à son accomplissement. Elle doit finalement être sous le contrôle d'une stratégie explicite.

## 3.5 Deux conséquences essentielles

### 3.5.1 Les robots réels et leurs environnements ne sont pas modélisables

Il ressort de la description des environnements réels ainsi que des difficultés propres aux perceptions une conséquence essentielle pour l'acquisition de comportements. Dans des conditions réalistes **les environnements des robots ne sont pas modélisables**, à moins que l'on ne modélise le monde de façon très restrictive en vue d'une tâche précise comme dans le cas par exemple d'un robot joueur de hockey sur table (cf. paragraphe 2.4.1). Vouloir modéliser complètement ces environnements s'apparenterait à répéter par exemple la tentative de Douglas Lenat dont le système Cyc devait contenir de manière massive toutes les informations nécessaires à la constitution du sens commun d'un système artificiel [Crevier, 1993]. Le coût d'une modélisation de l'environnement qui soit complète et opérationnelle est prohibitif.

Comme l'indique R. Brooks, il est nécessaire pour un agent autonome de se reposer non sur un modèle du monde mais sur les éléments perçus du monde réel : *A situated agent must respond in a timely fashion to its input. Modelling the world completely under these conditions can be computationally challenging. But a world in which it is situated also provides some*

*continuity to the agent. That continuity can be relied upon, so that the agent can use its perception of the world instead of an objective world model. The representational primitives that are useful then change quite dramatically from those in traditional Artificial Intelligence. The key idea from situatedness is : **The world is its own best model.***[Brooks, 1991a].

### 3.5.2 Les robots réels et leurs environnements ne sont pas simulables

Les caractéristiques d'un environnement réel rendent la réalisation de simulateurs réalistes extrêmement difficiles. La construction d'un simulateur est une tâche complexe qui doit être recommencée en partie pour chaque environnement particulier. L'impossibilité théorique de disposer d'un simulateur provient du fait que l'on ne dispose pas d'un modèle du monde opérationnel et complet. L'impossibilité pratique provient, quand à elle, du fait que l'on ne reportera pas dans l'environnement simulé tous les éléments de l'environnement réel susceptibles d'influer sur le robot. Dès que l'on fait appel à la vision, plutôt qu'à des senseurs de type distance/direction, les difficultés sont encore accrues. A l'inverse, *"Real world is for free"* [Brooks, 1991a].

Quelle que soit la stratégie utilisée, à partir de lois physique modélisées ou en intégrant des données issues de mesures réelles, des artefacts spécifiques à la simulation apparaissent. Comme l'indique R. Brooks [Brooks and Mataric, 1993] la reproduction quasi-parfaite d'une séquence d'action, la possibilité d'actions vraiment symétriques les unes des autres (durant un aller-retour par exemple), ou la simultanéité parfaite elle aussi de deux événements, sont propres aux simulateurs. Ulrich Nehmzow a par exemple montré expérimentalement [Nehmzow, 2001] l'émergence de comportements chaotiques dans les situations réelles : des comportements programmés très simples (suivi de mur, évitement d'obstacle) pour des environnements minimaux (arène rectangulaire), réalisés d'abord en simulation, puis en situation réelle, montraient une sensibilité aux conditions initiales beaucoup plus grande que dans les simulations (évaluation par estimation du premier exposant de Lyapunov).

Du fait de leur complexité et de leur coût de réalisation, la plupart des simulateurs de robots (Nomadic, ActiveMedia, Webbots, EasyBot etc.) n'implémentent que des modèles de bruit de type gaussien. Il est donc délicat de conclure qu'un système est robuste ou non après expérimentation sur ces simulateurs. Les simulateurs les plus élaborés proposent parfois

une vision simulée semblable à celle des jeux vidéo (via des moteurs 3D tels qu'OpenGL par exemple). Il est cependant fréquent, en recherche robotique, d'effectuer une phase préparatoire sur un simulateur puis de passer à une application réelle après adaptation des perceptions. Cette démarche peut convenir pour des comportements très génériques tels que l'évitement d'obstacles, mais pour des comportements situés dépendants d'éléments concrets, le saut en complexité est trop important. Certains travaux [Jakobi *et al.*, 1995] ont montré que, sous réserve d'une modélisation "précautionneuse" et avec un bruit adapté, des comportements simples obtenus par évolution en simulation fonctionnaient correctement dans la réalité. Mais cette approche est coûteuse et incertaine, puisqu'elle consiste d'abord à intégrer l'information sur le monde et les capteurs en les modélisant dans le simulateur puis à tenter de la retransmettre par évolution à un robot.

La simulation a une utilité certaine pour l'étude et l'élaboration de mécanismes comportementaux générique en permettant de se focaliser sur certains aspects de la relation agent/environnement. Elle ne permet pas, cependant, d'obtenir directement des comportements adaptés utilisables en situation réelle tels que ceux qui nous occupent ici (cf. chapitre 15). L'impossibilité pratique et théorique d'utiliser des simulateurs interdit d'employer les techniques qui nécessiteraient des stratégie d'essais/erreurs à grande échelle. L'acquisition de comportements doit être effectuée dans le temps du robot et de l'humain, si possible en environnement réel. *De l'impossibilité de modéliser et de simuler de façon satisfaisante il ressort la nécessité d'expérimenter directement dans les environnements auxquels sont destinés les robots.* De ce fait, des techniques comme les méthodes évolutionnistes ou l'apprentissage par renforcement souvent évoquées en robotique ne sont pas adaptés à ce type d'expérimentations.

## Chapitre 4

# Les difficultés propres à l'acquisition de comportements

### 4.1 Introduction

Un utilisateur ou un concepteur ayant pour objectif qu'un robot se comporte d'une manière donnée dans des circonstances précises est en fait confronté au *problème de la transmission de ses intentions*. Il lui faut pouvoir exprimer ce qui est important, ce qu'il faut faire et ce qu'il ne faut pas faire dans certaines conditions ou encore ce qui est utile et ce qui ne l'est pas. Il peut souhaiter que le comportement attendu *émerge* à partir des quelques consignes qu'il aura fixées, mais il ne souhaite pas qu'émerge n'importe quoi. Son problème n'est pas l'étude de tout ce qui peut émerger lors de l'interaction d'un robot avec son environnement mais plutôt l'acquisition de comportements utilisables ultérieurement. Il veut également pouvoir garder le contrôle car le robot ne doit pas devenir dangereux. De plus, le concepteur évolue éventuellement dans le même environnement que le robot, environnement auquel il est habitué et où il sait comment se comporter dans telle ou telle circonstance pratique. Plutôt que de commencer *ab initio*, il veut pouvoir réutiliser sa propre expérience. Il s'agit bien alors de trouver les moyens de cristalliser ou de synthétiser un certain nombre de ses *intentions* dans les comportements d'un système mécanique en tirant bénéfice de son *expérience*.

Après ce qui a été dit dans les chapitres 2 et 3 sur les perceptions et l'environnement il ressort que la principale difficulté provient du fait que nous ne disposons pas d'un modèle du

monde du robot réel et que le concepteur doit spécifier les comportements sans le support d'une description de l'environnement.

Pour transmettre un comportement à un robot trois voies principales sont envisageables :

- **Par programmation.**
- **Par apprentissage téléologique.**
- **Par apprentissage empirique.**

## 4.2 La programmation des comportements

On peut, par instruction directe, formuler des comportements en les programmant, à la condition que ces comportement puissent être exprimés au moyens d'informations clairement déductibles des perceptions. Ainsi des comportements classiques tels que l'évitement d'obstacle, le suivi de mur, le suivi d'objet coloré ou le désengagement peuvent être exprimés en maintenant des contraintes simples sur des variables telles que la distance et direction au plus proche obstacle, le nombre de pixels de la couleur de l'objet considéré ou encore la direction la moins obstruée.

Des comportements élémentaires peuvent être représentés par des programmes simples tel que le comportement *avoid* ci-dessous, extrait du contrôleur du robot Toto de M. Mataric [Mataric, 1992] ou encore sous la forme de schémas moteurs traduisant des contraintes spatiales [Arkin., 1989]. Ces composants sont ensuite combinables dans une architecture de sélection de l'action.

```
(defbehavior avoid
  (cond (( and (<= ( sonar 1 or 2) safe-zone)
              (<= ( sonar 3 or 4) safe-zone)) (turn left))
        ((<= (sonar 3 or 4) safe-zone) (turn right))))
```

Les comportements programmés peuvent bien sûr être bien plus complexes que celui indiqué ci-dessus, mais les limites de cette approche sont vite atteintes. Dès que le comportement ne peut plus être exprimé avec des invariants stables ou des perceptions fiables, sa programmation devient hasardeuse voire inaccessible. Les éléments nécessaires ne sont plus disponibles de

Comportement	invariant exprimé à partir de la mesure de :
Evitement d'obstacle	distance et direction de l'obstacle le plus proche
Suivi de mur	distance à droite ou à gauche
Suivi d'objet coloré	poind et centre de masse des pixels de couleur c
Désengagement	direction du secteur angulaire autour du robot le moins obstrué.

TAB. 4.1: Conception d'un comportement à partir d'un invariant simple

manière robuste dans les perceptions et *le comportement est hors de la portée d'une spécification explicite*. De plus, la modification et l'adaptation des comportements à des changements de l'environnement demande l'intervention d'un concepteur expert.

### 4.3 L'acquisition par l'apprentissage

L'apprentissage est envisagé par les roboticiens comme un moyen pour remédier aux difficultés rencontrées lors de la spécification explicite des comportements. Un panorama des approches les plus fréquemment utilisées est donné dans [Connel and Mahadevan, 1993] et [Arkin, 1998]. Les techniques de l'Apprentissage Artificiel y prennent une place croissante à l'instar des autres champs de l'IA, en raison de la plus grande *adaptabilité*<sup>11</sup> qu'elle confèrent aux systèmes qui les mettent en oeuvre.

Comme l'indique Jean Sallantin, l'apprentissage se caractérise toujours par une interaction [Sallantin, 1994]. En considérant la voie selon laquelle s'établit cette interaction, et selon laquelle le concepteur transmet ses intentions, on peut regrouper les méthodes d'apprentissage de comportements en deux grandes classes : celle des méthodes téléologiques et celle des méthodes empiriques.

#### 4.3.1 Transmission téléologique des intentions

Je regroupe dans la classe des approches téléologiques celles reposant sur une *formulation explicite, non plus du comportement complet, mais du but assigné au comportement*. Cette formulation explicite synthétique (par le moyen d'une récompense, d'une fonction d'évaluation) agit de manière téléologique sur la conformation finale du comportement. Ainsi, dans ces méthodes, on cherche à minimiser le nombre de contacts aux objets si l'on veut apprendre

<sup>11</sup>Le terme adaptativité correspondant plutôt aux capacités d'un robot s'adaptant dynamiquement à un environnement changeant

un comportement d'évitement, on maximise la durée de présence dans une zone éclairée pour un comportement phototaxique, on prend en compte la distance parcourue minimum dans un comportement de navigation, etc.

L'approche la plus utilisée est celle de *l'apprentissage par renforcement* [Kaelbling *et al.*, 1996; Mahadevan and Connell, 1992]. L'apprentissage par renforcement permet de déterminer par *essais/erreurs* la politique de contrôle d'un système autonome en cherchant à maximiser un signal de renforcement provenant de l'environnement. Le Q-Learning [Sutton, 1984] permet de prendre en compte des renforcements (récompenses) différés dans le temps.

Les *méthodes évolutionnistes* reprennent sous diverses formes le principe de la sélection naturelle et de la mutation/croisement des matériaux phénotypiques pour *générer et évaluer* des comportements. Parmi ces méthodes on trouve les *stratégies évolutionnistes* [Rechenberg, 1973; Schwefel, 1975], les *Algorithmes Génétiques* [Holland, 1975] et la *Programmation Génétique* [Koza, 1994]. Finalement, les *systèmes de classeurs* [Booker *et al.*, 1989] combinent l'approche évolutionniste et celle par renforcement.

Dans les systèmes robotiques autonomes ces méthodes sont souvent associées à des architectures à base de réseaux de neurones. Par exemple, dans le système de D. Floreano and F. Mondada, un robot Khepera apprend un comportement de navigation et d'évitement d'obstacles. Un algorithme génétique est utilisé pour sélectionner le réseau de neurones approprié. Le but du comportement est formulé explicitement et de manière très synthétique via la fonction d'évaluation (*fitness*) [Floreano and Mondada, 1994].

Ces méthodes ont permis d'obtenir des résultats très impressionnants *mais qui restent surtout cantonnés aux environnements simulés ou bien de type Arène*. Ceci en raison des deux principales difficultés qu'elles rencontrent :

1. Essais/erreurs : ces approches utilisent une stratégie par essais et erreurs qui impose pour converger d'évaluer de nombreuses variantes d'un comportement. Dans le cas d'un environnement réel dangereux (au moins pour le robot), cette approche n'est pas possible et l'apprentissage doit s'effectuer dans un cadre totalement contrôlé. Il n'est par exemple pas envisageable d'effectuer de nombreux essais d'accostage à une station de recharge,

ceux-ci risquant de mettre le robot rapidement hors d'usage. De plus ce grand nombre d'essais prend un temps beaucoup trop important en situation réelle.

2. Spécification : Les approches téléologiques nécessitent de spécifier une fonction de coût (fitness, reward) explicite et évaluable. Comme la programmation directe des comportements, cette spécification doit pouvoir s'appuyer sur des éléments extraits des perceptions (distance parcourue, durée à cet endroit, nombre de passages ici, etc.). Elle imposent donc de disposer d'un modèle du monde déjà très élaboré. La définition de cette fonction requiert les compétences d'un expert qui saura traduire l'objectif général en une fonction de coût utilisable par un système d'apprentissage particulier.

### 4.3.2 Transmission empirique des intentions

Les approche empiriques sont celles qui *s'appuient principalement sur l'exploitation d'expériences réalisées en situations réelles*<sup>12</sup> et qui ne nécessitent aucune spécification directe ou indirecte. Selon les termes usuels de l'Apprentissage Artificiel, elles correspondent aux méthodes d'apprentissage supervisé, dont un exemple type pour la robotique est le système ALVINN de D.A. Pomerlau [Pomerlau, 1993; Pomerlau, 1997].

Ces méthodes ont pour principal avantage de placer robot et concepteur directement dans l'environnement réel et exploitent les traces d'interactions se produisant entre le robot et l'environnement ainsi qu'entre le concepteur et le robot. Cependant leur succès dépend de leur capacité à tirer profit *d'expériences ponctuelles* qui ne sont que des *variantes* d'un comportement plus général. Le système ALVIN doit ainsi produire un comportement efficace de suivi de route par un véhicule autonome à partir de sessions de pilotage par un humain. En d'autres termes, ce succès dépend de leur capacité à *généraliser* à partir de ces expériences. Compte-tenu de l'intérêt de ces méthodes pour le problème que je traite ici, je reviendrai sur celles-ci dans le chapitre 6.

## 4.4 Biais

On peut considérer que l'apprentissage de comportements est un problème d'apprentissage parmi d'autres et qu'il bénéficiera des progrès des mécanismes d'apprentissage généraux et de

---

<sup>12</sup>Le terme empirique ici souligne l'importance prépondérante de l'expérience concrète, principalement introduit par la philosophie anglaise des XVIIe et XVIIIe siècles (John Locke, David Hume, George Berkeley).

leur aptitude à généraliser à partir de données concrètes. Mais il a été démontré [Wolpert and Macready, 1997] qu'il n'existe pas d'algorithmes d'apprentissage générique qui soit supérieur aux autres quel que soit le problème traité <sup>13</sup>.

Un système d'apprentissage est donc dépendant non seulement de l'algorithme d'apprentissage sur lequel il repose, mais aussi de l'ensemble des informations annexes qui lui permettent de prendre en compte le problème. Ces données annexes sont traditionnellement appelées *biais*. Tom Mitchell définit un biais comme étant " *any basis for choosing one generalization over another, other than strict consistency with the observed training sequence*" [Mitchell, 1980] et indique bien que vouloir apprendre indépendamment de tout biais n'a pas de sens [Cornuéjols and Miclet, 2002].

Dans le cas d'un robot, les biais peuvent par exemple porter sur la structuration de l'espace d'action, sur les perceptions disponibles et leur forme, sur les connaissances a-priori intégrées dans le système. Ils correspondent aux *prédispositions structurelles* que l'on met en oeuvre dans un système pour rendre possible l'apprentissage. Le problème de l'acquisition de comportements est lui aussi confronté à ce fait, et pas plus là qu'ailleurs il n'est possible de faire l'économie de l'étude de ces prédispositions structurelles.

## 4.5 Education

Les comportements utiles à apprendre sont souvent ceux pour lesquels il est *difficile de disposer d'une mesure objective utilisable*. En effet, pour beaucoup de comportements, l'intention du concepteur ne s'exprime plus dans un espace mesurable dans lequel il s'agirait de maximiser ou minimiser une fonction de coût (comme c'est le cas le plus souvent pour les méthodes téléologiques). De manière caricaturale on n'imagine pas qu'un robot apprenne à ramener des yaourts du frigidaire en utilisant comme seule information le nombre de yaourts tombés à terre.

En l'absence d'une telle mesure objective le seul autre critère disponible est la correspondance entre l'intention du concepteur et ce que réalise le robot. Pour que l'intention du concepteur

---

<sup>13</sup>"For any (optimization) algorithm any elevated performance over one class of problems is exactly paid for in performance over another class" (Wolpert)

---

prise sous cette forme soit transmissible au robot il faut que s'instaure *un processus interactif et incrémental entre le concepteur et le robot qui permette, étape par étape, de confirmer et amplifier la correspondance entre intention et réalisation*. Au cours de ce processus qui a l'apparence d'un processus *éducatif* traditionnel entre humains, le concepteur/tuteur doit pouvoir insister sur des points qui ont été mal appris, atténuer un comportement divergent, étendre ou reprendre un comportement déjà appris. En fait, ce n'est pas seulement l'intention du concepteur, mais l'intention du groupe humain concepteurs/utilisateurs qui est à prendre en compte sur le long terme. Pour être réellement utilisable, un système d'acquisition de comportements doit fournir les moyens de cette prise en compte.

A supposer qu'une telle "pédagogie" soit possible, elle se heurte au fait que concepteur et robot ne voient pas le monde de la même manière, leurs perceptions et aptitudes étant radicalement différentes. Cette différence de points de vues grève la capacité du tuteur à montrer ce qui compte au robot et il faut trouver les moyens d'homogénéiser les deux points de vue au travers des modalités d'apprentissage.

#### 4.5.1 Durée

Le temps de l'humain n'est pas celui de l'ordinateur et ceci a quelques conséquences pratiques. On ne pourra pas demander à un utilisateur de passer un temps trop important à produire de nombreux exemples dans le cadre d'un apprentissage supervisé. De plus cela lui sera certainement très pénible s'il doit recommencer à chaque fois que l'environnement change ( par exemple l'ammeublement d'une pièce). Il ne sera pas là non plus pour systématiquement donner le renforcement attendu ou pour surveiller pendant des heures un robot qui apprend par essais et erreurs. Autrement dit, le processus 'éducatif' doit rester compatible avec ce que sont prêts à concéder les utilisateurs.

#### 4.5.2 Intégration

Finalement la question se pose de savoir comment peuvent être employés les comportements que l'on aura su transmettre à un robot. Supposons que l'on ait pu lui apprendre : un comportement d'accostage à une station de recharge, plusieurs comportements de passage (passage d'une porte, entrée dans un ascenseur, etc..), des comportements de vérification (s'approcher d'un dispositif d'incendie par exemple), un comportement pour récupérer un objet, un autre pour le déposer. On souhaitera ensuite combiner ces composants de base dans

une application complète.

Cette *intégration* d'un comportement dans une application nécessite d'abord de pouvoir le déclencher dans les *contextes* appropriés et donc de savoir identifier ces contextes (on ne déclenche pas un comportement d'ouverture de porte devant une table); elle nécessite encore de pouvoir le sélectionner parmi d'autres, ou de savoir lorsqu'il est terminé. Enfin, l'intelligibilité des représentations utilisées pour les comportements facilitera toutes les opérations indispensables de monitoring et de maintenance (par l'utilisateur ou le concepteur). On verra, chapitre 12, comment ces questions sont abordées dans le système que je propose.

## 4.6 Conclusion

En résumé, l'acquisition de comportements est rendue difficile par l'absence de modèle du monde et la distance entre simulation et monde réel. Elle doit par ailleurs prendre en compte les contraintes temporelles imposées par les limites matérielles et humaines. La spécification directe (par programmation) ou indirecte (par une méthode téléologique) ne peut convenir que pour une partie des comportements. Plus le comportement est dépendant des spécificités physiques et sociales (règles, desiderata des utilisateurs) de l'environnement, plus l'apprentissage doit s'effectuer *in-situ* en l'absence de mesure objective. Les méthodes empiriques permettant de mettre en oeuvre un processus éducatif dans lequel un concepteur/tuteur construit incrémentalement un comportement sont de bons candidats pour l'acquisition de composants comportementaux situés.

# Chapitre 5

## Incidences sur les systèmes d'acquisition de comportements

### 5.1 Introduction

Les difficultés décrites dans les précédents chapitres peuvent être récapitulées sous la forme de quelques problèmes synthétiques auxquels sera confronté tout système d'acquisition de comportements.

- **Problème 1 : Sans modèle du monde, sur quoi s'appuyer pour spécifier les comportements ?**

En général, nous ne pouvons pas disposer d'un modèle de l'environnement dans lequel le robot doit évoluer. Seuls quelques aspects peuvent en être modélisés, souvent à un coût important.

- **Problème 2 : Comment déterminer ce qu'il faut percevoir pour un comportement donné ?**

Le robot doit percevoir ce qui est nécessaire pour réaliser une action et il doit prendre en compte la nature bruitée et altérée des perceptions dans l'environnement réel.

- **Problème 3 : Comment adapter le robot à son environnement réel ?**

Sous peine d'être inopérant, voire même endommagé, le robot doit être adapté à son environnement réel. Le robot n'évolue pas dans un monde virtuel mais le plus souvent dans des environnement régis par des contraintes physiques et sociales nombreuses.

- **Problème 4 : Comment intégrer l'humain dans la boucle, comment celui-ci peut-il formuler ses intentions ?**

Les moyens d'apprentissage doivent être conçus dans l'idée que les humains ne sont pas prêts à montrer 1000 exemples à chaque fois qu'ils changent la peinture de leurs murs et qu'ils n'ont pas non plus un temps infini pour récompenser ou punir un système si la machine à café à été changée de place.

- **Problème 5 : Comment intégrer les comportements acquis à une application complète ?**

Un comportement ne forme pas en soi une application utile. On doit pouvoir l'intégrer à une application complète, combinant éventuellement plusieurs comportements. Cette application doit avoir le moyen de sélectionner et contrôler les comportements.

## 5.2 Propriétés requises

Les problèmes ci-dessus délimitent un ensemble de propriétés essentielles que doit selon moi viser un système pour l'acquisition de comportements d'usage de l'environnement.

1. **EDUC : Propriétés relatives à l' Education**

Permettre à un utilisateur humain (à priori non expert) de formuler ses intentions et d'enseigner des comportement variés à un robot au travers d'un processus éducatif qui soit compatible avec les contraintes pratiques usuelles (temporelles et spatiales).

2. **ENV : Propriétés relatives à l'Environnement**

S'appliquer à des environnement réels variés sans nécessiter de modèle préalable de ceux-

ci.

### 3. PERC : Propriétés relatives aux Perceptions

Déterminer automatiquement le support perceptuel d'un comportement, c'est-à-dire les percepts à partir desquels seront enclenchées les actions. Encoder de manière robuste la relation perception/action en tenant compte des altérations qui se produisent normalement dans les environnements réels.

### 4. INTEGR : Propriétés relatives à l'Intégration

Donner la possibilité d'intégrer les comportements à une application robotique complète en fournissant les interfaces nécessaires à la sélection, au déclenchement, et à la terminaison des comportements unitaires.

## 5.3 Incidence sur les moyens utilisables

Répondre à ces propriétés nécessite d'aborder plusieurs difficultés simultanément ; il peut paraître plus profitable de les traiter chacune séparément en tentant de résoudre des problèmes jouets ou du moins plus élémentaires. Mais il faudra que l'on soit capable ensuite de relier ces problèmes jouets aux problèmes réels sans faire trop de pré-suppositions sur ce qu'il est véritablement possible de percevoir, de modéliser et de mesurer. Je pense pour ma part que cette convergence de difficultés forme au contraire la spécificité des problèmes de robotique autonome et leur intérêt. *On retrouvera une convergence similaire dans les applications de plus en plus nombreuses impliquant de manière générale des agents et environnement réels - protection de l'environnement, surveillance, prévision/simulation.*

Les propriétés ci-dessus ont une incidence directe sur les méthodes et techniques que l'on peut envisager. L'absence de modèle interdit d'utiliser une approche instructive, donc la programmation ou la planification s'avèrent a priori inutilisables à ce niveau. L'absence ou la rareté de symboles pertinents interdit d'utiliser les méthodes d'apprentissage symbolique. L'absence de simulateur réaliste interdit des stratégies d'essais/erreurs nécessitant pour converger un très grand nombre d'essais. Les approches téléologiques ; telles les stratégies évolutionniste ou l'apprentissage par renforcement, sont de ce fait inapplicables : pour le cas d'un comportement

d'accostage à une station de recharge, on ne fera pas essayer au robot les centaines de tentatives nécessaires pour obtenir par évolution ou renforcement le comportement attendu. Les contraintes de l'utilisateur humain ainsi que la taille des espaces de recherche interdisent d'utiliser des méthodes nécessitant de nombreux exemples (ie : Perceptron-Multi-Couches, fonctions à base radiale) et des phases d'apprentissages longues (ie : Support Vector Machine).

## 5.4 Principale hypothèse

Trois acteurs interviennent dans le processus d'acquisition de comportement : le robot, l'utilisateur humain et l'environnement. A partir de ce constat, et en suivant les propriétés

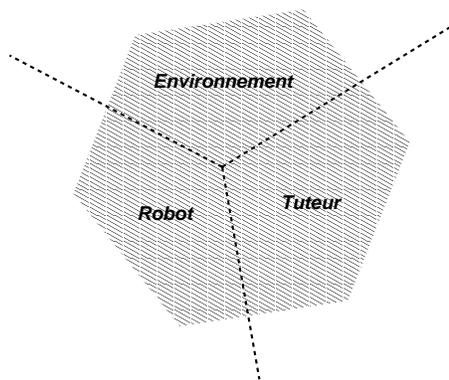


FIG. 5.1: Les trois acteurs intervenant lors de l'acquisition d'un comportement

du précédent paragraphe, ma principale hypothèse est celle d'un ***apprentissage interactif situé*** selon laquelle : ***l'acquisition d'un comportement résulte des interactions entre le robot et son tuteur humain dans l'environnement de destination.***

Dans les chapitres qui suivent, je présenterai les travaux existants compatibles avec cette hypothèse générale, puis je proposerai une solution reposant sur une représentation distribuée.

## Chapitre 6

# Approches empiriques et interactives de l'apprentissage de comportements

### 6.1 Introduction

Ce chapitre fait le point sur les différents travaux qui se situent dans le cadre de l'hypothèse principale d'*apprentissage interactif situé* énoncée dans le chapitre précédent. Tous ces travaux ont comme point commun de mettre en relation, en vue de l'apprentissage, un tuteur et un robot dans un environnement réel. Les travaux qui reposent sur une interaction homme/robot en vue de l'apprentissage d'un comportement, ou d'une aptitude, sont relativement peu nombreux. Beaucoup moins nombreux que ceux utilisant par exemple l'apprentissage par renforcement dans des univers simulés. Cependant, si l'on examine les articles parus dans les principales conférences de robotique ces toutes dernières années (ICRA,IROS,...), on note un intérêt croissant pour ce type de démarche (à IROS et ICAR particulièrement). Plus généralement une place de plus en plus importante est consacrée aux travaux portant sur le problème de la relation homme/robot en conditions réalistes [Dautenhahn, 1995],[Breazeal and Scassellati, 1999].

Les différents travaux entrant dans le cadre de l'hypothèse générale peuvent être regroupés sous le terme de méthodes d'apprentissage empiriques interactionnelles.

On peut répartir ces travaux dans trois catégories principales :

1. L'apprentissage par démonstration.
2. L'apprentissage par imitation.
3. Les méthodes hybrides reposant sur l'apprentissage par renforcement.

### **Apprentissage par démonstration**

Sous ce terme peuvent être rangés différents travaux d'apprentissage en robotique se réclamant de l'apprentissage par démonstration, de l'apprentissage par l'exemple, de l'apprentissage à partir de cas, ou du "teaching by showing" ou encore du "learning by watching" et, plus généralement, l'apprentissage supervisé. L'apprentissage par démonstration peut être rapproché historiquement et dans son esprit de la *programmation par démonstration* [Cypher, 1993], approche proposant de composer des programmes (via l'interface utilisateur) à partir de suite d'opérations effectuées par un utilisateur.

La plupart des systèmes reprennent l'approche inductive et symbolique provenant de la programmation par démonstration et ont pour but essentiel de produire des plans synthétisant les démonstrations. Ces approches sont de ce fait très dépendantes d'une modélisation correcte de l'environnement. Les travaux relevant d'un apprentissage par démonstrations plutôt inductif sont principalement ceux de H.Friedrich et R.Dillman [Friedrich and Dillman, 1995],[Friedrich and Kaiser, 1995]. D'autres travaux reposent plutôt sur des techniques connexionnistes ou plus généralement numériques que symboliques. C'est le cas des systèmes présenté dans [Pomerlau, 1997],[Moore *et al.*, 1995][Atkeson and Schall, 1997],[Bentivegna and Atkeson, 2000]. En règle générale, la plupart des travaux se réclamant de l'apprentissage par démonstration n'ont pas une approche située et prennent peu en compte l'environnement. De plus, hormis le système ALVINN de D.A. Pomerlau, ces approches ne sont pas appliquées à des robots mobiles autonomes.

### **Apprentissage par imitation**

Dans l'apprentissage par imitation un robot apprend en observant un autre agent, ce dernier pouvant être un autre robot ou un tuteur humain. L'imitation suppose d'établir une relation de correspondance entre les mouvements observés et les mouvements à réaliser. En plus

des problèmes partagés avec l'apprentissage par démonstration, l'imitation est donc confrontée à la tâche très difficile de l'identification explicite ou non des mouvements de l'autre agent et de leur transposition dans le point de vue du robot. On trouve une revue générale de ces travaux dans l'article de P. Bakker et Y. Kuniyoshi, [Bakker and Kuniyoshi, 1996].

Une définition très générale de l'imitation proposée par ces auteurs ("*il y a imitation lorsqu'un agent apprend un comportement en observant la réalisation de ce comportement par un professeur*") peut tout à fait permettre d'assimiler apprentissage par démonstration et par imitation dans un cadre plus vaste. Mais elle élimine les problèmes essentiels de l'imitation : la correspondance et la transposition.

Les travaux relevant de l'apprentissage par imitation peuvent être répartis dans deux catégories :

1. Imitation forte : travaux s'intéressant au problème de l'imitation *per se*, en connexion avec les données de la psychologie et de la biologie.
2. Imitation faible : travaux concevant l'imitation plus comme un moyen technique pour l'apprentissage, indépendamment des modèles proposés pour l'homme et l'animal.

Les travaux de la première catégorie sont souvent inspirés des recherches sur la psychologie de l'enfant comme ceux de J. Piaget [Piaget, 1977] ou les travaux plus récents de Meltzoff [Meltzoff and Moore, 1992] ainsi que des recherches en neurophysiologie établissant une correspondance entre actions observées et actions réalisées<sup>14</sup>. On trouve dans cette catégorie les travaux de P. Gaussier [Gaussier *et al.*, 1997] et P. Andry [Andry *et al.*, 2001]. Ces travaux proposent des modèle d'imitation reposant sur la correspondances entre la dynamique de phénomènes externes perçus et celle des mécanismes de contrôle propre au robot, sans qu'il soit nécessaire de passer par une phase d'identification explicite des phénomènes à imiter. Les moyens pour capturer cette dynamiques reposent principalement sur un type de réseaux de neurones récurrents spécifique (composant PerAc) capable d'apprendre et de prédire une séquence temporelle.

On retrouve l'usage d'un autre type de réseau récurrent dans les travaux de A. Billard [Billard

---

<sup>14</sup>Les neurones miroirs mis en évidence chez les singes par G. Rizzolatti [Rizzolatti *et al.*, 1996] sont des neurones du cortex pré-moteur qui sont actifs lors de la réalisation d'une action et lorsque cette même action est observée chez un congénère.

and G.Hayes, 1999] [Billard, 1999] que l'on peut situer entre les deux catégories. Les travaux relevant de la deuxième catégorie proposent d'utiliser l'imitation comme un moyen rapide et direct pour transmettre un comportement. Pour ce faire, ils reposent le plus souvent sur la reconnaissance et le séquençage de *primitives perception/action* comme c'est le cas dans les travaux de G.Hayes, J.Demiris ou M.Mataric, [Demiris and Hayes, 1996] [Hayes and Demiris, 1994] [Demiris and Mataric, 1998] ou encore sur un modèle très précis de l'environnement qu'il s'agit alors de confronter aux données sensorielles courantes [Kuniyoshi *et al.*, 1994]. Les capacités d'apprentissage de ces modèles sont d'emblée cadrées par les primitives disponibles ou l'environnement décrit. Ils correspondent plus à un apprentissage par démonstration rendu interactif qu'à une imitation véritable.

### Méthodes hybrides reposant sur l'apprentissage par renforcement

Les méthodes hybrides décrites ici sont celles qui combinent les méthodes classiques d'apprentissage par renforcement [Sutton, 1984],[Mahadevan and Connell, 1992],[Kaelbling *et al.*, 1995] avec un mécanisme incrémental géré par un tuteur pour rendre possible l'apprentissage en conditions réelles. Dans son cadre général, l'apprentissage par renforcement consiste à trouver une politique <sup>15</sup> qui maximise la somme des récompenses reçues au cours du temps par le robot. L'apprentissage par renforcement suppose donc bien dans son principe que l'agent apprenant interagisse avec son environnement par le biais de récompenses. Cependant, en pratique, la difficulté est de formuler et de rendre disponible ces récompenses pour le robot, celui-ci devant être capable de prendre en compte des récompenses différées par rapport à l'action courante. La construction d'une politique optimale suppose l'examen par le robot de très nombreuses situations, ce qui est difficilement réalisable. Le propos des méthodes hybrides est de tenter de remédier à ces difficultés pour rendre possible l'apprentissage en situation réelle.

Le système décrit par Minoru Asada [Asada *et al.*, 1996] rentre dans cette catégorie. Ce système est utilisé pour apprendre à un robot à frapper dans un ballon dans le cadre de la compétition RoboCup. Le robot dispose d'un système perceptuel dédié lui permettant de décrire la balle et le but à l'aide de quelques attributs de position et de taille, et apprend en effectuant des tentatives dans de nombreuses situations. Pour remédier au problème de

---

<sup>15</sup>Une politique  $Q(tat, action)$  associe une valeur à chacun des couples (état, action) accessibles.

la récompense différée, Asada propose d'apprendre d'abord à partir de missions faciles issues de la tâche visée <sup>16</sup> et organisées par ordre de complexité croissante (celles-ci démarrant de plus en plus loin du but). Les 'missions faciles' correspondent à un sous-ensemble des états connexes de l'espace d'états total.

Le système proposé par M. Dorigo et M. Colombetti [Dorigo and Colombetti, 1994] [Dorigo and Colombetti, 1998] repose sur l'utilisation d'un système de classeurs <sup>17</sup> pour permettre à un robot situé d'apprendre des comportements élémentaires dans un environnement de type arène. L'apprentissage par renforcement est ici aussi incrémental, et décomposé en trois phases successives : bébé, jeune et adulte. Dans la phase "bébé", le tuteur récompense chaque action positivement ou négativement. Dans la phase "jeune", le robot raffine par lui-même en recevant des récompenses différées de l'environnement. Dans la phase "adulte", il n'apprend que plus que lorsque qu'une baisse de performance est détectée par un système de monitoring.

Parmi les approches empiriques hybrides n'utilisant pas l'apprentissage par renforcement, on citera les travaux de O. Lebeltel et P. Bessière [Lebeltel *et al.*, 2000] reposant sur l'inférence bayésienne et combinant une phase de spécification ou de programmation (définition des variables et distributions jointes utilisées) à une phase d'apprentissage permettant la caractérisation statistique des variables utilisées.

## 6.2 Description de systèmes empiriques

### 6.2.1 D.A. Pomerlau - apprentissage supervisé numérique.

Le système ALVINN (Autonomous Land Vehicle In a Neural Network ) proposé par Dean A. Pomerlau [Pomerlau, 1997] est un exemple bien connu d'apprentissage supervisé. Ce système a été conçu pour permettre à un véhicule (un van) de suivre une route dépourvue de trafic de manière autonome. Le fonctionnement autonome est obtenu après une session d'apprentissage pendant laquelle le véhicule est piloté par un humain. ALVINN réussit à réaliser cette tâche à partir d'un seul perceptron multicouches entraîné par rétro-propagation. Le ré-

---

<sup>16</sup>LEM : Learning from Easy Missions

<sup>17</sup>Les Learning Classifier Systems [Booker *et al.*, 1989] représentent un contrôleur à l'aide de règles perceptions-actions appelées classeurs. Celui-ci utilise le renforcement pour apprendre le poids des règles et un algorithme génétique pour générer de nouvelles règles.

seau de neurones a pour entrée une image vidéo de basse résolution et pour sortie la commande de braquage du véhicule.

Dans l'article [Pomerlau, 1993] D. A. Pomerlau insiste sur deux points particulièrement importants. Premièrement, il lui a été nécessaire de générer des exemples supplémentaires à partir des exemples issus des sessions de pilotage. Les exemples initiaux ne pouvaient en effet pas contenir de situations dans lesquelles le pilote corrigeait la trajectoire et la diversité des situations contenues dans l'ensemble d'apprentissage était insuffisante. Des exemples ont donc été créés par transformation des images des exemples de l'ensemble d'apprentissage. Cette transformation était nécessairement très dépendante de la tâche considérée. Tel que le décrit l'auteur, le façonnage de l'ensemble d'apprentissage a fait partie intégrante de la pédagogie.

Deuxièmement, les expérimentations sur le système ALVINN ont montré que pour des situations très différentes (route à une voie, à deux voies, chemin) il était plus efficace d'entraîner plusieurs réseaux distincts plutôt qu'un seul. Ceci illustre la dépendance au contexte d'apprentissage des techniques utilisées, et plus généralement le problème que constitue la prise en compte des différents contextes.

Par ailleurs, comme le remarque T.Ziemke [Ziemke, 1999b], ALVINN ne peut prendre en compte le *perceptual aliasing* (ou ambiguïté perceptuelle), c'est à dire le fait que des perceptions identiques puissent correspondre à des situations différentes. Par exemple, un véhicule qui s'approche puis s'éloigne d'un obstacle aura des perceptions visuelles identiques dans deux situations différentes (s'il n'y a pas rémanence de la vision). Dans ALVINN, ceci est dû au fait que le système utilise un réseau de neurones de type feed-forward et ne dispose pas de mémoire de son état.

### 6.2.2 G.Atkeson and S.Schaal - Apprentissage par démonstration

Le système décrit par G.Atkeson and S. Schaal [Atkeson and Schall, 1997] s'attaque à l'apprentissage par démonstration d'une tâche particulièrement difficile. Un pendule doté d'une tige rigide est suspendu à un bras robotisé. Initialement il pend à la verticale au-dessous de la main du robot. Le système doit apprendre à faire passer le pendule à la verticale au-dessus de la main du robot et à le maintenir ainsi en équilibre. Les démonstrations du tuteur humain sont observées par un système de vision stéréo produisant l'information de position du pendule et de la main du tuteur. Le même système de vision est utilisé pour le fonctionnement autonome.

Atkeson et Schaal indiquent qu'il n'est pas possible d'imiter directement les mouvements humains car le pendule est trop sensible aux légères différences de mouvements qui ne manquent pas de se produire du fait de l'imprécision des perceptions et des différences de dynamique entre la main humaine et la main du robot. Ils proposent d'apprendre à planifier les mouvements du robot en apprenant une politique de contrôle par renforcement. Les démonstrations du tuteur humain sont utilisées pour calculer les récompenses et pour *amorcer* le processus d'apprentissage lors des premiers essais.

Le détail du calcul des récompenses, la nature des perceptions prises en compte, la part de modélisation rendent cette approche très dépendante de la tâche considérée. Cependant, l'intérêt de ce système réside principalement dans la proposition d'une approche hybride qui combine renforcement, modélisation partielle et initialisation d'un processus de planification par démonstration.

### 6.2.3 J.Hayes and J Demiris - Apprentissage par Imitation

Dans leur article "A Robot Controller Using Learning by Imitation", J.Hayes et J. Demiris expérimentent l'apprentissage d'un parcours de labyrinthe [Hayes and Demiris, 1994]. Un robot Elève apprend par imitation en suivant et en observant un robot Tuteur, ce dernier programmé pour accomplir un parcours fixe. Dans une première phase, le robot Elève suit le robot Tuteur en maintenant sa distance. Il observe, via une caméra, les mouvements du Tuteur et il associe les changements de direction observés à ses propres perceptions courantes. Plus précisément, il associe sa perception des obstacles (via des capteurs infrarouge) à la rotation qu'il faut effectuer pour maintenir son Tuteur au centre du champ de vision. La perception des mouvements du robot Tuteur est facilitée par le fait que ce dernier est doté de deux rangées de diodes électroluminescentes, situées à l'avant et à l'arrière. L'orientation du robot Tuteur est ainsi simplement obtenue à partir de l'image vidéo seuillée et binarisée pour ne laisser apparaître que la rangée de diodes dans ses différentes orientations.

Dans une deuxième phase, le robot Elève reproduit seul le parcours. Il se fie alors à ses perceptions courantes et reproduit les changements de directions mémorisés. L'association perception/mouvement est représentée par des règles ou primitive élémentaires. Dans des travaux ultérieurs, J. Hayes et A. Billard ont proposé une architecture connexionniste pour des tâches d'imitation similaires (cf. architecture Drama à base de neurones récurrents [Billard

and G.Hayes, 1999]).

Cet exemple d'apprentissage par imitation met en oeuvre des mécanismes relativement simples et montre qu'il est possible d'obtenir par ces moyens une imitation élémentaire d'un autre robot. Cependant de nombreux biais perceptuels sont employés qui restreignent beaucoup les actions et perception possibles : le robot tuteur se réduit pour l'élève à une rangée de diode orientée, les actions disponibles correspondent aux seuls quarts de tours nécessaires pour sortir du labyrinthe.

#### 6.2.4 M.N. Nicolescu, M. Mataric : Apprentissage de tâches

Dans le système proposé par M.N. Nicolescu et M. Mataric [Nicolescu and Mataric, 2001], le but visé est de produire la représentation d'une tâche après une démonstration. Le robot est doté d'un comportement de recherche d'objet  $Track(color, angle, distance)$  bien défini et supporté par un télémètre laser et une caméra couleur. Dans les tâches traitées, le robot doit par exemple effectuer un slalom entre des plots colorés (en fonction des aptitudes du système perceptif) ou visiter ces plots les uns après les autres dans un ordre donné.

Durant la démonstration, le système observe les intervalles temporels durant lesquels des instances particulières du comportement  $Track$  sont utilisées ( $Track(Green, 179, 468)$  puis  $Track(Orange, 12, 590)$ , etc.). Puis il construit un automate dont les noeuds seront ces instances particulières, en déterminant les pré-conditions et leurs contraintes. Ce système montre bien que lorsque l'on sait précisément ce que l'on cherche (ici quelques objets colorés) et que ce que l'on cherche est facile à percevoir, il est ensuite relativement aisé de construire une représentation de la tâche visée (ici sous forme d'automate). Autrement dit, la difficulté réside bien plus dans la détermination des éléments perceptuels nécessaires à la tâche que dans sa représentation.

### 6.3 Conclusion

Les approches empiriques interactionnelles pour l'apprentissage rencontrent un intérêt croissant depuis quelques années et s'illustrent par des travaux très variés. Cependant il ne ressort pas de ce panorama un ensemble précis de techniques, pas plus qu'un cadre conceptuel hormis pour le cas spécifique de l'imitation forte. Dans la majorité des cas, il s'agit avant tout d'apprendre et de représenter des *séquences d'actions* en utilisant des dispositifs perceptuels

limités et adaptés aux tâches envisagées. Ce dernier point diminue radicalement la complexité intrinsèque des problèmes.

A mon avis, l'adaptation de la perception est sous-estimée ou ne constitue pas, en tout cas, un aspect important dans la plupart de ces systèmes. Comme je l'ai déjà évoqué dans les chapitres précédents, les aptitudes perceptuelles d'un système conditionnent d'emblée ses capacités d'apprentissage dans le monde réel et ne peuvent pas être considérées comme subalternes. La partie suivante présentera les hypothèses structurantes, le cadre ainsi que les mécanismes détaillés du modèle que je propose pour l'acquisition de comportements. Ce modèle est résolument orienté vers la captation de la relation perception/action au niveau le plus bas.



Deuxième partie

Un modèle pour l'acquisition de  
comportements



## Avant-propos

Cette partie est principalement consacrée à la description du modèle, appelé **MPL**, que je propose pour la représentation et l'apprentissage de comportements situés. Avant de le détailler, je présente dans le chapitre 7 les hypothèses qui m'ont permis de le structurer. Le chapitre 8 rassemble et précise les éléments constituant le cadre du modèle. Le détail du modèle proprement dit est donné dans les chapitres 9, 10 et 11. Les moyens employés pour la perception sont finalement présentés dans le chapitre 13.



## Chapitre 7

# Elaboration d'un modèle d'acquisition de comportements

### 7.1 Introduction

L'hypothèse générale énoncée dans le chapitre 5 conduit à chercher un modèle permettant l'acquisition de comportements sous la conduite d'un utilisateur humain. Pour construire celui-ci, il est indispensable de partir de ce qu'il est réellement possible de percevoir dans l'environnement réel. Pour cette raison, je privilégie une approche *bottom-up*, partant des perceptions élémentaires pour aboutir aux représentations de plus haut niveau. Le modèle doit être autant que possible 'minimal' et ne pas reposer sur un agencement complexe de méthodes d'apprentissage et de reconnaissance de formes, agencement qui serait finalement difficilement analysable et très dépendant de l'application envisagée.

Ce chapitre présente plusieurs hypothèses, portant plus particulièrement sur la perception, et qui seront utilisées pour élaborer la structure du modèle d'acquisition de comportements.

### 7.2 Pistes pour construire le modèle

Les hypothèses structurant le modèle sont en partie inspirées par des disciplines situées dans le champ de la biologie telles que la neurophysiologie et l'éthologie. Le propos n'est pas ici de valider un modèle biologique ou de se référer systématiquement à la biologie pour justifier les choix retenus ; mon objectif est la construction d'un artefact qui ne doit tirer sa justification que par les applications qu'il permet d'envisager. Cependant les problèmes rencontrés par

la robotique autonome dans les environnement réels ont déjà trouvé leur solution dans le règne animal [Wilson, 1991]. Au moins à titre d'inspiration, plusieurs faits physiologiques et biologiques sont susceptibles d'influencer la structure des modèles, ou en tout cas de remettre en cause le modèle instructif que tout informaticien a traditionnellement à l'esprit.

### 7.2.1 Affordances

Les modèles des activités d'un agent dans son environnement proposés par l'éthologie et la psychologie ne se ramènent pas à l'application d'un plan reposant sur une représentation abstraite du monde. Ces disciplines invoquent au contraire un couplage fort entre perceptions et possibilités d'actions instanciées dans l'environnement. Cette vision des choses peut être résumée par une phrase de Maurice Merleau-Ponty, l'un des philosophe ayant contribué à donner un nouveau cadre à la perception et à l'action : *"Le monde visible et celui de mes projets moteurs sont des partie totales du même Etre"* [Merleau-Ponty, 1964].

Dans ses travaux sur l'écologie de la perception, J.J. Gibson à proposé, par l'intermédiaire de sa *théorie des Affordances* [Gibson, 1986], de considérer que l'environnement se présente à l'animal comme un ensemble d'opportunités d'actions qu'il perçoit directement<sup>18</sup>. Un agent observant son environnement y perçoit des "affordances" en relation directe avec ses caractéristiques spécifiques (sa taille, les mouvements qu'il peut faire, etc.), plutôt que les propriétés abstraites et physique des objets. J.J Gibson ne considère pas la perception comme une simple entrée mais comme un organe à part entière. *"The affordance of the environment are what it offers the animal, what it provides or furnishes either for good or ill. If a terrestrial surface is nearly horizontal, nearly flat, and sufficiently extended and if its substance is rigid (relative to the weight of the animal) then the surface affords support, and we call it a substratum, ground or floor. It is stand-on-able, permitting an upright posture for quadrupeds and bipeds. It is therefore walk-on-able and run over-able. ....The four properties listed - horizontal, flat, extended and rigid - would be physical properties of surfaces if they were measured with the scales and standard unit used in physics. As an affordance of support for a species of animal however they have to be measured relative to the animal. They are unique for that animal. They are not just abstract physical properties."*

---

<sup>18</sup>On retrouve cette approche dans le champ des sciences cognitives avec D. Norman [Norman, 1988]et W.H Warren [Warren, 1995] en particulier pour les conséquences qu'elle aurait sur le design des objets de la vie courante.

Selon Gibson, la niche écologique d'un animal est un ensemble d'affordances. Il suggère que les affordances peuvent être en partie déduites des déformations du flot optique (*ambient optic array*) dans lequel est plongé l'animal - c'est-à-dire du flot visuel perçu par l'animal en mouvement dans son environnement.

La théorie de Gibson ne mène pas à un modèle opérationnel utilisable pour les robot autonomes - les déformations du flot optique ont été étudiées dans de nombreux travaux sans que ressortent clairement un modèle pour les comportements. Cependant, elle peut être considérée comme un cadre général plaçant le noeud du problème à la frontière agent/environnement. Elle suppose d'abord que les éléments de l'environnement ne doivent être considérés que relativement au robot, sans qu'intervienne de mesures physiques objectives - on retrouve cette idée sous la forme des représentations déictiques proposée par Philip Agre<sup>19</sup>. La théorie des affordances suggère que le robot doit pouvoir percevoir ce qu'il peut faire - ou ce qu'il a le droit de faire si on la transpose dans le cas d'un environnement social - sans passer par un processus interprétatif; l'environnement perçu fournit directement un support pour l'action.

### 7.2.2 Population

Plusieurs expériences de neurophysiologie montrent que des processus à l'oeuvre chez l'animal ou l'homme reposent sur l'activité conjuguée d'un grand nombre d'entités opérant de concert, plutôt que sur des blocs fonctionnels bien délimités, hiérarchisés et séquencés.

A. Berthoz [Berthoz, 1997] rapporte une expérience réalisée par A.P. Georgopoulos [Georgopoulos *et al.*, 1982] montrant que c'est l'activité d'une grande *population de neurones* qui est associée à une action. Dans cette expérience on demande à un singe de pointer du doigt dans plusieurs directions et on relève que certains neurones du cortex moteur déchargent plus que d'autres pour une direction donnée. On enregistre ensuite l'activité de plusieurs de ces neurones lorsque le singe doit pointer dans une seule direction précise. Si on fait la moyenne de toute ces décharges on trouve alors que le *vecteur moyen de la population* est très précisément orienté dans la direction du mouvement du doigt vers la cible.

---

<sup>19</sup>Extrait de [Agre, 1988] "*Deictic representation means individuating things in the world not objectively (independently of the agent's location or heading or project or attitudes) but rather indexically (in terms of their relation to the agent) and functionally (in terms of the role they play in the agent's ongoing projects)*".

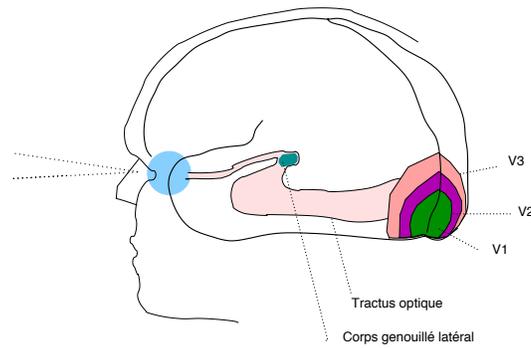


FIG. 7.1: Les trois aires primaires du cortex visuel : V1, V2 et V3.

On retrouve cette notion de population de neurones dans le système perceptuel. L'expérience du neurophysiologiste Roger Tootell est à ce titre particulièrement intéressante [Tootell *et al.*, 1982]. R. Tootell a stimulé le champ visuel d'un oeil de macaque avec une image représentant une cible constituée de rayons et de cercles concentriques; il avait préalablement anesthésié l'animal durant 45 minutes et lui avait injecté un produit marquant les neurones actifs. Après l'expérience, ceux-ci sont visualisables sur une plaque photographique (fig. 7.2 extraite de [Changeux and Ricoeur, 2000]). On constate que l'on retrouve, dans une coupe de l'aire visuelle primaire V1 du cortex cérébral, le tracé des rayons et cercles concentriques de la cible. Cette homologie de forme entre un stimulus visuel géométrique et l'état d'activité d'une zone cérébrale montre que, dans ce cas, *l'organisation rétinotopique* du stimulus est conservée assez loin dans le cortex cérébral (via le tractus optique et le corps genouillé latéral). Alors que l'on pourrait s'attendre à trouver très vite des neurones codant une information de nature hiérarchique et plus synthétique - comme le feraient les neurones intermédiaires de la couche cachée d'un perceptron- on retrouve au contraire les éléments discrets du champs perceptuel. Cette correspondance topographique est aussi décrite par Gerald M. Edelman ([Edelman, 1992] page 41).

### 7.2.3 Perceptions élémentaires

La neurobiologiste Margaret Livingstone [Livingstone, 1989] indique que ces populations de neurones du système visuel se répartissent nettement en trois voies codant trois propriétés simples : la couleur (contraste de chrominance), la forme élémentaire (contraste de luminance) et la direction du mouvement [Poggio and Koch, 1989]. De plus, cette ségrégation subsiste jusque dans l'aire visuelle V2 (cf. figure 7.1). Notre culture nous porte à penser que notre sys-

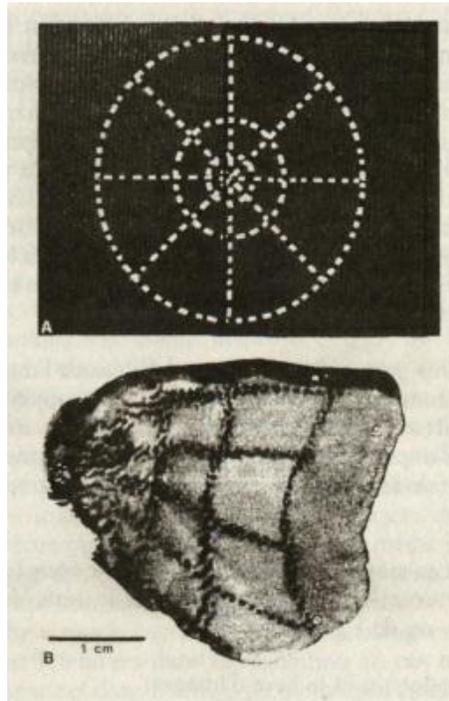


FIG. 7.2: Homologie de forme entre stimulus et zones activées. Dans l'expérience de R. Tootel, un macaque à été exposé au stimulus visuel représentant une cible, après l'expérience on retrouve le stimulus dans une coupe de l'aire V1 du cortex visuel.

tème perceptuel a pour principale tâche d'associer des objets abstraits aux stimuli sensoriels, mais on constate - au moins jusqu'à l'aire V2 - que seules des propriétés perceptuelles élémentaires sont relevées, qu'il n'y a pas systématiquement de hiérarchisation et que la population de neurones conserve l'organisation topologique du stimulus initial.

Par ailleurs l'éthologie animale a depuis longtemps fait la preuve que des comportements complexes chez les animaux peuvent être déclenchés par des traits perceptuels très simples. L'éthologue N. Tinbergen a montré par exemple [Tinbergen *et al.*, 1942] que le stimulus déclencheur du comportement de poursuite sexuelle chez le papillon de jour *Eumenis Semele* peut être qualifié précisément en terme de couleur, taille, forme élémentaire et mouvement. L'éthologie animale décrit de nombreux comportements supportés/déclenchés par des traits perceptuels simples [Lorenz, 1978; Ruwet, 1975].

## 7.3 Hypothèses structurant le modèle

La section précédente ne prétendait pas donner un panorama de la neurobiologie. Elle rappelait simplement quelques faits qui sont pour la plupart bien connus. Ces faits et les constats des chapitres 2, 3 et 4 m'ont mené à structurer le modèle d'acquisition à partir des quatre hypothèses suivantes :

1. Hypothèse de distribution
2. Hypothèse de tessellation
3. Hypothèse du couplage perception/action
4. Hypothèse de façonnage

### 7.3.1 Hypothèse de distribution

Cette première hypothèse consiste à concevoir *un modèle distribué dont le fonctionnement repose sur une population d'entités élémentaires*, ces entités étant identiques et non hiérarchisées. Elle s'inspirent des faits relevés par la neurobiologie et de l'idée que les activités cérébrales sont supportées par les activités conjuguées de nombreux systèmes simples [Minsky, 1988]. Ce point de vue est à la croisée de deux importants paradigmes de l'IA que sont le connexionisme et les systèmes multi-agents [Ferber, 1995].

### 7.3.2 Hypothèse de Tessellation (micro-perception)

Cette hypothèse consiste à n'utiliser que des *micro-percepts* élémentaires et aucun macro-percepts. Un micro-percept porte sur une petite partie fixe du champ perceptuel, il s'agit par exemple d'un pixel d'une couleur donnée à une position fixée dans le référentiel rétinien. Un macro-percept correspond à une description de plus haut niveau, (ie : une forme rectangulaire de couleur bleue) , qui implique un processus de *reconnaissance* plus complexe et qui est plutôt défini par les attributs descriptifs d'un objet structuré (couleur, forme géométrique). Il est difficile à reconnaître du fait de l'altération des perceptions.

Le fait de doter un robot d'un vocabulaire de micro-percept a moins de conséquences sur ce qu'il est possible d'apprendre que de doter ce même robot d'un vocabulaire de macro-percept. Par analogie, lorsque l'on regarde les tableaux pointilliste de Seurat, les premiers Matisse ou encore les mosaïques antiques, on constate qu'un vocabulaire limité d'éléments discrets permet

de restituer une grande variété de situations. Le robot percevra le monde comme un ensemble de tesselles d'une mosaïque et non comme une liste d'objets à reconnaître.

### 7.3.3 Hypothèse du couplage perception/action et de concomitance

Cette hypothèse consiste à coupler fortement dans le modèle (et donc dans les structures de données le supportant) la représentation de la perception à celle de l'action. Pour les phases d'apprentissage elle consiste à considérer les événements *concomitants* se produisant sur le plan de la perception et sur le plan de l'action. Elle suppose de considérer perception et action comme étant reliée par une relation de concomitance plutôt que de causalité.

### 7.3.4 Hypothèse du façonnage (shaping)

Cette dernière hypothèse porte sur la stratégie d'apprentissage. Selon cette hypothèse, c'est le tuteur humain qui façonne le comportement de manière incrémentale par essais successifs. Cette approche qui intègre l'utilisateur humain dans la boucle veut compenser l'absence d'une fonction de coût servant de guide à l'apprentissage. La plupart des méthodes d'apprentissage supposent de disposer d'une fonction de coût explicite mais, pour le problème de l'acquisition de comportements dans un environnement réel, une telle fonction de coût n'a pas véritablement de sens. D'une part l'environnement n'est pas modélisable et il est impossible de se reporter aux éléments d'un modèle du monde pour construire une mesure explicite. D'autre part, le tuteur humain ne peut montrer que des *facettes* de ce qu'il faut faire au travers d'exemples et d'interactions limitées, et il n'y a aucun intérêt à apprendre strictement ces exemples. La seule fonction de coût véritablement utile ici consiste plutôt en le fait que tuteur humain soit satisfait ou non à l'usage.



## Chapitre 8

# Cadre pour le modèle d'acquisition de comportements

### 8.1 Introduction

Ce chapitre précise le cadre de l'acquisition des comportements. Il décrit ce que j'appellerai *le robot élémentaire*, c'est-à-dire l'abstraction au travers de laquelle sera vue le robot. Il présente ensuite les démonstrations résultant de l'interaction entre le robot et son tuteur. Pour servir de point de repère le problème de l'acquisition est d'abord examiné dans le contexte et avec les outils de l'apprentissage supervisé.

### 8.2 Le robot élémentaire (RE)

Pour concevoir un modèle d'acquisition de comportements, nous avons besoin de nous appuyer sur un robot réel et sur une description précise de son fonctionnement. Le robot décrit doit rester suffisamment simple pour permettre une modélisation claire ne dépendant pas d'éléments spécifiques relatifs à tel ou tel aspect matériel. En effet, quand les travaux en robotique requièrent des configurations trop particulières il est impossible de les comparer et distinguer les apports respectifs de l'ingénierie et des solutions proposées ; Il est important de définir un modèle de fonctionnement suffisamment générique pour espérer à terme comparer les solutions et les modèles d'apprentissage. Cependant la configuration choisie doit aussi rester suffisamment riche pour permettre de prendre en compte une grande variété de situations et de se confronter aux difficultés résumées dans le chapitre 5.

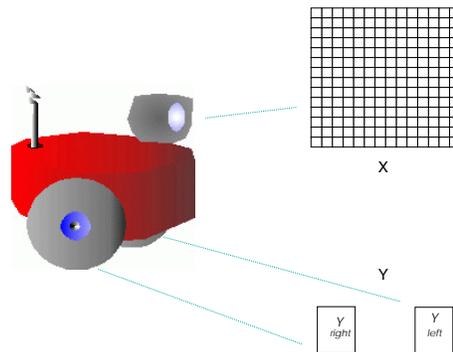


FIG. 8.1: Le robot élémentaire

Je définis ici un 'robot élémentaire' facilement transposable sur de nombreux robots réels et c'est à ce robot que je ferai référence dans la suite du texte. Le robot élémentaire est doté d'un senseur et de deux effecteurs. Le senseur est une caméra vidéo 2D (désormais peu coûteux et de plus en plus répandu), qui fournit une information riche et complexe et surtout apparaît adapté à une grande variété de situations<sup>20</sup>. Nous disposons nous-mêmes d'un senseur visuel relativement semblable. Cette relative similarité entre les perceptions du robot et celles des utilisateurs humains peut faciliter l'interprétation des actions du robot en permettant à un utilisateur humain de deviner plus facilement quel trait de l'environnement influence le robot à un moment donné. Par ailleurs, le fait d'utiliser un senseur produisant des données sous forme matricielle confère au robot élémentaire un caractère assez générique car les autres senseurs fournissent une information qui peut aussi être ramenée à un matrice 2D (les grilles d'occupation pour les senseurs de type distance/direction, le relevé d'un gradient pour les senseurs de type intensité direction).

Les effecteurs de ce robot élémentaire sont deux roues motrices, dont la vitesse de rotation est contrôlable séparément. Avec ce type de roue différentielles, le robot est presque holonome c'est-à-dire qu'il peut passer facilement d'une posture (position, orientation) à une autre sans devoir passer par des étapes intermédiaires (comme c'est le cas pour un robot hétéronome disposant par exemple de quatre roues).

Les actions que le robot peut effectuer sont visualisables dans un espace d'état à deux dimen-

<sup>20</sup>Dans la nature on trouve ce type de senseur chez une multitude d'espèce animales, elles mêmes adaptées à toutes sortes d'environnements. Les senseurs de type distance/direction quant à eux ne sont présents que chez les espèces beaucoup moins nombreuses capables d'écho-localisation (dauphin et chauve-souris, par exemple).

sions appelé espace d'action, présentant respectivement en abscisse et en ordonnée la vitesse de la roue gauche et celle de la roue droite. Les trajectoires continues dans cet espace correspondent à tous les mouvements réalisables par le robot (fig. 8.2).

Il serait possible de rajouter à la description du robot élémentaire de nombreux aspects tels que le détail de sa forme, son poids, un modèle de ses caractéristiques cinématiques ou celles de son champ visuel. Mais je choisis ici de ne pas intégrer ces informations parce qu'elles sont trop dépendantes d'un robot spécifique et *qu'elles peuvent être prises en compte implicitement au travers d'interactions réelles avec l'environnement.*

La robot élémentaire (fig. 8.1) est entièrement décrit par la définition de ses entrée/sorties :

- une matrice réelle  $X$  donnant la couleur de chaque point du champ visuel.
- un vecteur réel  $Y = [Y_{left}, Y_{right}]^T$  pour le contrôle des mouvements.

Ce modèle de fonctionnement ne nécessite aucune connaissance spécialisée en robotique. Il met en oeuvre des perceptions suffisamment complexes pour permettre de se confronter à toutes les difficultés énoncées dans le chapitre 5. Les problèmes de robotique sont particulièrement difficiles du fait des aspects mécaniques, électroniques et physiques des robots, mais je pense pour ma part que le véritable problème consistant à associer les 'bonnes actions' aux 'bonnes perceptions' reste entier y compris en utilisant un robot aussi simple que ce robot élémentaire. Posé en ces termes il concerne directement l'Intelligence Artificielle.

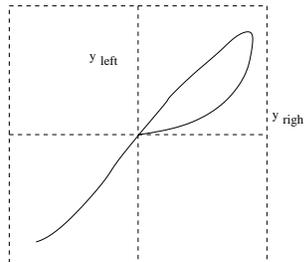


FIG. 8.2: Les mouvements du robot sont représentables par une courbe dans l'espace  $y_{left} \times y_{right}$

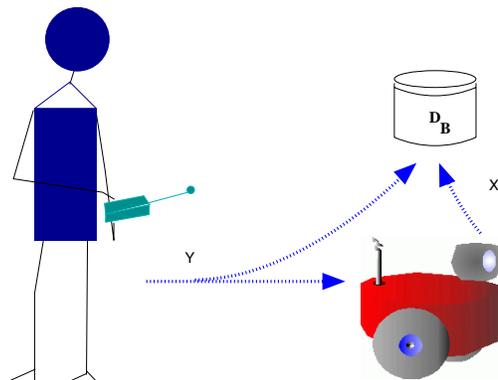


FIG. 8.3: Le robot télécommandé

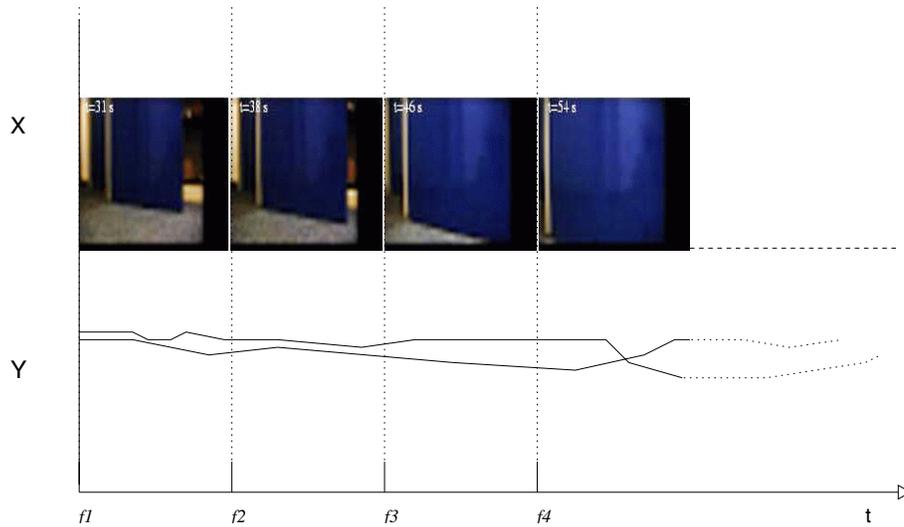
### 8.3 Apprentissage à partir de démonstrations

Si le tuteur souhaite que le robot apprenne un comportement tel que l'accostage à une station de recharge ou la sortie d'une pièce, sans lui donner aucune définition explicite (équationnelle ou programmé) il n'a d'autre moyens que de lui 'montrer' ou de lui 'dire' ce qu'il faut faire, de lui en faire la démonstration de manière empirique. Pour cela la façon la plus directe est de prendre le contrôle du robot et de le manipuler à distance. Il n'est pas nécessaire d'avoir recours à un processus d'imitation qui impliquerait que le robot soit capable de déceler et de transposer les actions d'un autre agent dans son propre espace d'action. D'ailleurs, dans le cas d'une imitation véritable, le tuteur occulterait en partie le champ visuel du robot et rendrait la perception plus complexe encore. L'imitation, qui constitue une problématique en soi, n'est pas une technique indispensable pour l'acquisition de ce type comportements.

Le tuteur a besoin simplement d'un mécanisme de contrôle à distance qui lui permette de produire tous les mouvements possibles du robot, c'est-à-dire toutes les trajectoires dans l'espace d'action. Il doit pouvoir se substituer temporairement au mécanisme de contrôle autonome du robot.

Une session pendant laquelle le tuteur contrôle ainsi le robot est appelée une *démonstration* [Friedrich and Kaiser, 1995] et consiste en l'enregistrement au fil du temps des perceptions et actions du robot. Une démonstration  $d$  est donc une suite de  $m$  frames, nom donné aux couples, associant pour un même instant  $t$  les données sensorielles  $X_t$  à la consigne du tuteur,  $Y_t$  (eq. 8.1).

$$d = \{f_1, f_2, \dots, f_m\} = \{\{X_1, Y_1\}, \{X_2, Y_2\}, \dots, \{X_m, Y_m\}\} \quad (8.1)$$

FIG. 8.4: Les frames d'une démonstration  $d$ 

Lors d'une même démonstration le tuteur ne peut montrer qu'une *variante* du comportement voulu - la figure 8.5 présente trois variantes d'un même comportement. Chaque variante correspond à la réalisation du comportement depuis un point donné, dans une posture donnée et dans des conditions environnementales particulières (d'illumination par exemple). Il est nécessaire de fournir un mécanisme qui permettent d'obtenir un *comportement synthétique* à partir de plusieurs démonstrations, c'est-à-dire à partir de plusieurs réalisations concrètes du comportement imaginé par le tuteur. Les démonstrations sont a priori peu nombreuses car elles proviennent d'un utilisateur humain.

L'ensemble des  $l$  démonstrations correspondant à un comportement  $B$  est noté  $D_B$  (eq 8.2).

$$D_B = \{d_1, d_2, \dots, d_l\} \quad (8.2)$$

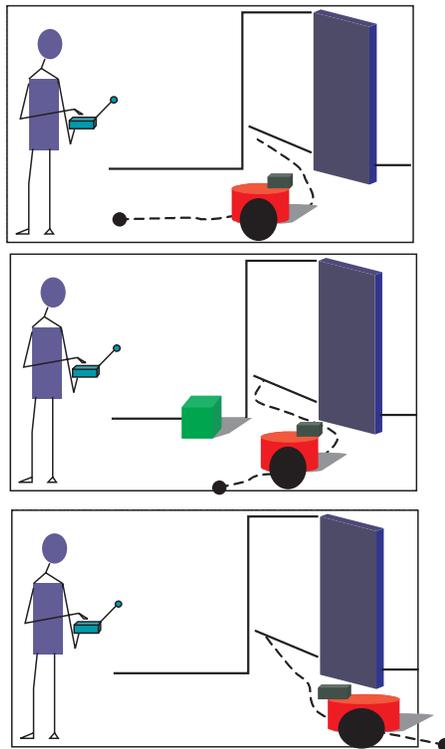


FIG. 8.5: Trois démonstrations pour l'apprentissage d'un comportement de passage de porte. Elles représentent chacune une variante du comportement désiré.

## 8.4 L'acquisition dans le cadre de l'apprentissage supervisé

Avant de présenter **MPL**, il convient de se poser une question : Le problème de l'acquisition d'un comportement à partir de l'ensemble de démonstrations  $D_B$  peut-il être replacé dans le cadre de l'apprentissage supervisé et résolu directement par une technique de ce domaine ? Un problème d'apprentissage supervisé est souvent décrit [Vapnik, 1999] [Haykin, 1999] comme ayant trois composantes :

1. Un environnement générant des vecteurs aléatoires  $X \in R^n$  tirés d'une densité de probabilité  $F(X)$  inconnue.
2. Un superviseur retournant une valeur de sortie  $Y$  pour chaque vecteur d'entrée  $X$  selon une distribution conditionnelle  $F(Y|X)$  elle aussi inconnue.
3. Une machine apprenante capable d'implémenter un ensemble de fonctions  $f(X, \alpha)$ ,  $\alpha \in \Lambda$ ,  $\Lambda$  étant un ensemble de paramètres.

Le problème d'apprendre à partir des exemples devient alors celui de choisir dans l'ensemble de fonctions  $f(X, \alpha)$ , celle qui approxime le mieux les réponses du superviseur. Il s'agit d'un problème de régression si  $Y$  est prend des valeurs réelles et d'un problème de classification si  $Y$  ne prend que des valeurs discrètes.

Si on applique ce cadre au cas de l'apprentissage de comportement, l'environnement est alors celui que le robot perçoit au travers de ses senseurs, le superviseur est la consigne  $Y$  indiquée par le tuteur. Puisque  $Y$  est réel le problème s'apparente à un problème de régression.

Le choix de la meilleure approximation repose sur l'utilisation d'une fonction de coût  $L(Y, f(X, \alpha))$  qui mesure la perte entre la réponse  $Y$  du superviseur et la réponse  $f(X, \alpha)$  pour une même entrée  $X$ . On cherche donc, parmi les fonctions implémentables, la fonction  $f(X, \alpha_0)$  qui minimise le *risque* pour toutes les paires  $(X, Y)$  en fonction de leur distribution (eq. 8.3).

$$R(\alpha) = \int L(X, f(X, \alpha)) dF(X, Y) \quad (8.3)$$

Cependant, comme dans tous les cas réels, la distribution  $F(X, Y)$  des paires  $(X, Y)$  (distribution de probabilité jointe sur  $X$  et  $Y$ ) est inconnue. Nous ne disposons que d'une connaissance indirecte de  $F(X, Y)$  au travers des exemples  $(X_1, Y_1), \dots, (X_n, Y_n)$ . Dans ce cas il reste possible de chercher  $\alpha$  qui minimise le *risque empirique* sur l'ensemble des exemples :

$$R_{emp}(\alpha) = \frac{1}{n} \sum L(Y_i, f(X_i, \alpha)) \quad (8.4)$$

Pour un problème de régression le critère de coût optimal est celui des moindres carrés. Et l'équation 8.4 devient :

$$R_{emp}(\alpha) = \frac{1}{n} \sum ((Y_i - f(X_i, \alpha))^2) \quad (8.5)$$

### 8.4.1 Expérimentation préliminaire

L'exemple type de machine apprenante pouvant être utilisée dans ce cadre est celui du Perceptron Multicouches (PMC). Ce type de réseau de neurones est un approximateur universel de fonctions et est susceptible de représenter toutes les fonctions souhaitées ; l'apprentissage du PMC est par exemple réalisé par rétro-propagation de l'erreur, ce qui correspond préci-

sement à minimiser le risque empirique. On trouvera une description détaillée du PMC dans [Mitchell, 1997] et [Bishop, 1995].

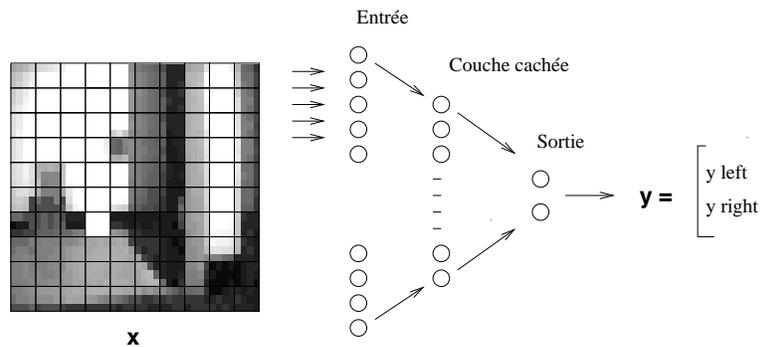


FIG. 8.6: Utilisation d'un perceptron multicouche pour l'apprentissage supervisé

On trouvera dans la section 15.3.3 les résultats d'une expérience préliminaire mettant en oeuvre un PMC pour l'apprentissage de comportement. A la suite de cette expérience il m'a été possible de faire plusieurs constatations :

1. l'apprentissage de ce type de comportements avec un PMC donne de mauvais résultats.
2. Le nombre d'exemples est trop restreint pour permettre d'obtenir un comportement satisfaisant.
3. L'apprentissage est beaucoup trop lent et est incompatible avec un usage interactif.
4. Les données issues de la phase d'apprentissage - ici les poids du réseau - sont impossibles à interpréter à posteriori.
5. Le réseau traite en permanence toutes les données perceptuelles de la même manière et il est impossible de prédire les effets particuliers d'une situation jamais vue. La réaction d'un robot utilisant un PMC peut être totalement imprévisible et éventuellement dangereuse.

Il ne m'a évidemment pas été possible dans le cadre de cette thèse d'essayer tous les mécanismes d'apprentissages pouvant être appliqués à un problème de régression de ce genre (ie. fonctions à base radiale, Support Vector Machines). Cependant la plupart des remarques précédentes peuvent leur être appliquées.

En plus du précédent constat, deux points plus fondamentaux indiquent que le cadre de l'apprentissage supervisé et de ses moyens n'est pas strictement celui qui convient à l'apprentissage de comportements.

Le premier point touche à la transmission de l'intention de l'utilisateur. En effet *l'intention de l'utilisateur, c'est-à-dire l'idée qu'il se fait du comportement à obtenir n'est pas résumable à l'association d'un critère d'erreur (ie. les moindres carrés) et d'un ensemble de démonstrations*. La difficulté tient à ce que le comportement tel que se le figure le tuteur n'est pas un objet mesurable, ainsi qu'au fait que les démonstrations sont peu nombreuses et ne couvrent pas toutes les situations. Finalement, seul l'avis du tuteur est déterminant. Celui-ci doit pouvoir amender/rectifier le comportement en faisant partie de la boucle d'apprentissage.

Le deuxième point touche au biais perceptuel introduit dans le système. Quel que soit le mécanisme envisagé, ses capacités d'apprentissage dépendront de la façon dont lui sont présentées les données, le biais utilisé lui permettant ou non de généraliser efficacement.<sup>21</sup>. Par conséquent l'étude des biais utilisables et en particulier des biais perceptuels est aussi importante que celle des mécanismes d'apprentissages génériques. On verra, au chapitre 11 qu'il est possible d'utiliser un mécanisme d'apprentissage très simple en association avec un biais perceptuel approprié.

---

<sup>21</sup>Cette idée a déjà été présentée dans la section 4.4. Elle est clairement exposée par T. Mitchell [Mitchell, 1980] *"If consistency with the training instance is taken as the sole determiner of appropriate generalizations, then a program can never make the inductive leap necessary to classify instance beyond those that it has observed. Only if the program has other source of information, or biases for choosing one generalization over the other, can it non-arbitrarily classify instance beyond those in the training set.... We use the term bias to refer to any basis for choosing one generalization over another, other than strict consistency with the training instances"*

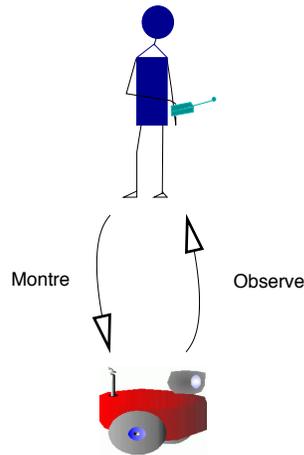


FIG. 8.7: Schéma général de la synthèse interactive de comportement

## 8.5 Schéma général du processus de synthèse

Le schéma idéal serait donc celui de la figure 8.7, schéma dans lequel le tuteur montre le comportement attendu et observe le résultat produit, le système chargé de l'apprentissage devant combiner deux opérations fondamentales :

- une opération de *synthèse* chargée de construire le comportement à partir de ce que montre le tuteur.
- une opération de *réalisation* prenant en charge la réalisation autonome du comportement.

Un schéma plus interactif (avec intervention de l'utilisateur au cours de la réalisation) sera abordé dans le chapitre 17. Mais il est nécessaire au préalable de disposer d'un modèle d'acquisition de comportements dans lequel la synthèse et la réalisation de comportement sont différées. Comme le montre la figure 8.8, le tuteur est intégré dans une boucle dans laquelle se succèdent les phases de synthèse à partir des démonstrations et les phases de réalisation autonomes. Pour que ce schéma soit réaliste il est essentiel que la phase de synthèse soit rapide et donc que la complexité en temps des algorithmes d'apprentissages utilisés soit faible. Les chapitres suivants présenteront le modèle de synthèse supervisée proposé, conçu pour permettre d'amorcer l'apprentissage en partant de dispositions perceptuelles et d'action minimales.

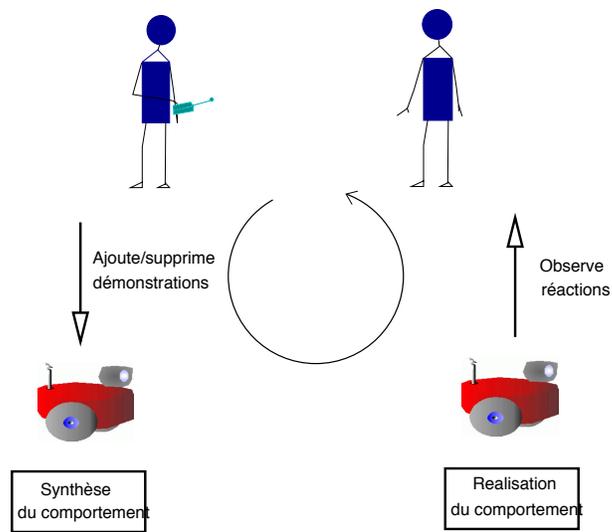


FIG. 8.8: Schéma général de la synthèse supervisée de comportement



## Chapitre 9

# MPL Modèle pour la synthèse supervisée de comportements

### 9.1 Introduction

Le modèle **MPL** <sup>22</sup> proposé reprend les hypothèses énoncées dans la section 7.1, c'est à dire les hypothèses de distribution, de tessellation, de concomitance et de façonnage. Il est conçu pour remédier aux désavantages des méthodes classiques d'apprentissage supervisé dans le cadre spécifique de l'apprentissage de comportements. En particulier les raisons qui rendent difficiles l'acquisition de comportements pour les méthodes numériques génériques sont le trop grand nombre d'exemples nécessaires, la lenteur du processus d'apprentissage et l'inintelligibilité des données produites. Le mécanisme proposé s'inspire de plusieurs faits relevant de la neurophysiologie et de la biologie plaidant pour une structure distribuée supportée par des entités élémentaires non hiérarchisées. Son fonctionnement est principalement de nature statistique.

Le but du modèle est d'obtenir un comportement synthétique  $B$  à partir d'un ensemble restreint  $D_B$  de démonstrations produites par le tuteur. Le modèle présenté est *un modèle à population* dans lequel un comportement synthétique  $B$  est représenté par un ensemble de cellules non hiérarchisées.

---

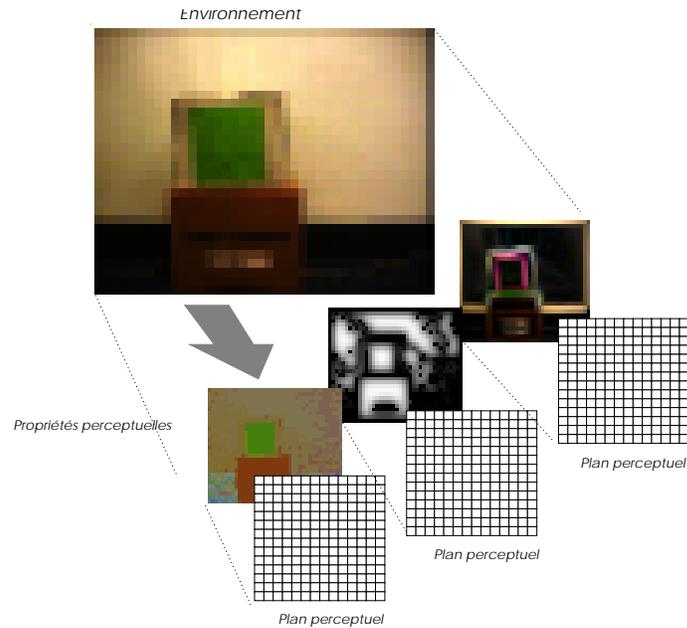
<sup>22</sup>Micro-Population Based Learning

Dans ce chapitre, je présente deux variations du modèle à population. Toutes les deux permettent de représenter, d'apprendre et d'utiliser un comportement  $B$ . La première variation est appelée *Modèle taxique*, la deuxième *Modèle à échantillons*. La première variante reprend directement les hypothèses. Elle est d'abord présentée parce qu'elle est plus intuitive et fournit un bon support pédagogique pour exposer et introduire le modèle à échantillons. Ce dernier repose sur des éléments moins intuitifs et tire les enseignements de l'application de la variante taxique pour fournir une solution beaucoup plus robuste.

## 9.2 Structure du modèle MPL

La structure commune aux deux variantes repose sur les éléments suivants :

1. un champ perceptuel
2. une population de cellules
3. un mécanisme de synthèse
4. un mécanisme de réalisation du comportement

FIG. 9.1: *Champ perceptuel*

### 9.2.1 Champ perceptuel et tesselles

Le robot perçoit son environnement au travers d'un champ perceptuel (CP) (fig. 9.1). Celui-ci correspond aux données sensorielles transformées et projetées sur un espace discret. Il est découpé en plusieurs plans correspondant chacun à une *propriété perceptuelle* unique, telles que la couleur ou la luminance. La projection des données sensorielles sur chaque plan du champ perceptuel conserve une *homologie de forme* avec le stimulus sensoriel initial. Ainsi, pour un stimulus visuel, on retrouvera l'organisation rétinotopique de l'image de départ dans les différent plans perceptuels.

Chaque plan perceptuel est un espace discret à deux dimensions dont les cases peuvent prendre une série de valeurs discrètes appelées *tessels* par analogie avec les tesselles d'une mosaïque (fig 9.2). Le tessel est l'unité d'information perceptuelle élémentaire, il est décrit par sa *position* dans un plan perceptuel donné et la valeur attendue de la propriété perceptuelle à cette position (eq. 9.1). *On peut considérer le tessel comme un pixel dans une image perceptuelle* (par exemple un point d'une couleur donnée à une position précise dans le plan des couleur du champ visuel ou encore l'intensité de la présence d'un obstacle dans une direction).

$$tessel \equiv (Couleur(position) = valeur\ discrete) \quad (9.1)$$



FIG. 9.2: *Le monde vu avec des tesselles - Mosaïque Romaine d'Utique en Tunisie (détail), datant probablement de la fin du IIIe siècle après J.C (Musée du Bardo). Un vocabulaire limité de tesselles permet de représenter une situation complexe.*

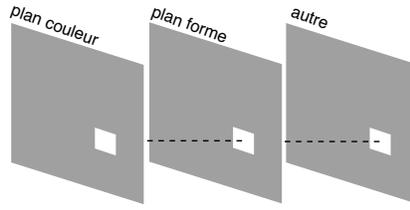
Comme on le verra plus en détail dans le chapitre 13, *c'est en structurant le champ perceptuel, c'est-à-dire en définissant les plans et les propriétés utilisées que le concepteur peut introduire un biais dans le modèle.*

Exemples de tessels :

- $tessel_a \equiv (Couleur(25, 30) = rougevif)$
- $tessel_b \equiv (Couleur(25, 38) = vertsombre)$
- $tessel_c \equiv (Distance(12, 1) = proche)$
- $tessel_d \equiv (Forme(2, 12) = 5)$

Dans la suite du chapitre on utilisera le champ perceptuel  $CP_1$  dont la structure est donnée ci-dessous (on reviendra sur la justification de cette structure dans le chapitre 13). Le champ perceptuel  $CP_1$  est constitué uniquement des deux plans *Couleur* et *Forme*, qui ont comme propriétés respectives :

- Le plan *Couleur* donne pour chaque point du champ visuel sa catégorie de couleur, représentée par une valeur discrète.
- Le plan *Forme* donne pour chaque point du champ visuel la catégorie de forme correspondant à la zone locale autour de ce point. Il s'agit plus d'une proto-forme que d'une

FIG. 9.3: *Micro-percept*

forme élaborée, elle correspondra à une transition, un changement de contraste. Cette catégorie est représentée par une valeur discrète.

La *conjonction* des tessels survenant simultanément dans les différents plans à une même position est appelée un *micro-percept* (fig. 9.3). Dans la suite du chapitre, seul ce type de configuration sera considéré et le champ perceptuel sera vu par les algorithmes comme un ensemble de micro-percepts.

### 9.2.2 Cellules

Le comportement  $B$  est représenté par un ensemble de  $n$  cellules, noté  $\mathcal{C}_B$ .

$$\mathcal{C}_B = \{cell_1, cell_2, \dots, cell_n\} \quad (9.2)$$

La cellule (fig. 9.4 et eq. 9.3) est l'élément central du modèle. Son rôle consiste à associer perception et action à un niveau de granularité très bas. Elle enregistre la relation de concomitance qui s'établit entre un percept et une action. Elle est constituée par la définition d'une conjonction de micro-percepts  $x_i$  (eq. 9.4) et d'une action  $y_i$  pondérée par un poids  $w_i$ .

$$cell_i \equiv (x_i, y_i, w_i) \quad (9.3)$$

$$x_i = mp_1^i \wedge mp_2^i \wedge \dots \wedge mp_p^i \quad (9.4)$$



FIG. 9.4: Une cellule vue comme un arc-réflexe entre un micro-percept et une action

Une cellule  $cell_i$  est *activée* quand tous les micro-percepts de  $x_i$  sont présent simultanément

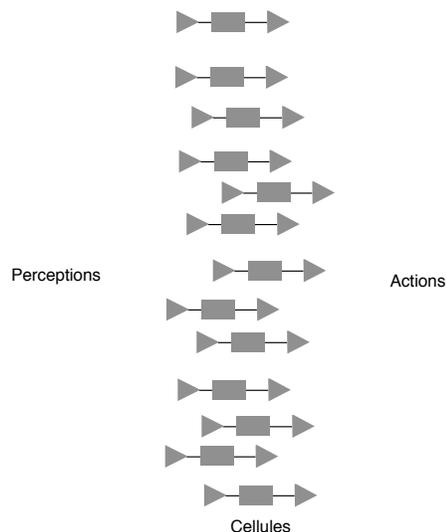


FIG. 9.5: La population de cellules relie perceptions et actions

dans le champ perceptuel (eq. 9.5).

$$active(cell_i) \equiv \forall mp_j^i \in x_i, mp_j^i \in CP \quad (9.5)$$

La partie action de la cellule représente l'action à réaliser lorsque la cellule est *active*. L'action est représentée par une instance du vecteur effecteur  $Y$ . Elle est pondérée par un gain représentant l'influence de cette cellule.

La cellule peut être considéré comme un micro arc-réflexe reliant un stimulus à une sortie ou encore comme une micro-règle de la forme « si  $mp_1^i \wedge mp_2^i \wedge \dots \wedge mp_p^i$  alors action  $y_i$  avec gain  $w_i$  ». La relation entre perception et action est donc représentée par un ensemble de cellules, chacune responsable d'un micro arc-réflexe (fig. 9.5). Dans l'esprit du modèle, les cellules sont nombreuses mais chacune ne contient que quelques micro-percepts.

## Support

Les données sensorielles  $X$  déterminent à tout moment, au travers des micro-percepts présents dans le champ perceptuel  $CP$ , un sous ensemble de cellules actives parmi les cellules appartenant à  $\mathcal{C}_B$  (eq. 9.6). Cet ensemble est appelé le support du comportement  $B$  pour  $X$  et noté  $\mathcal{S}_B(X)$  (fig. 9.6).

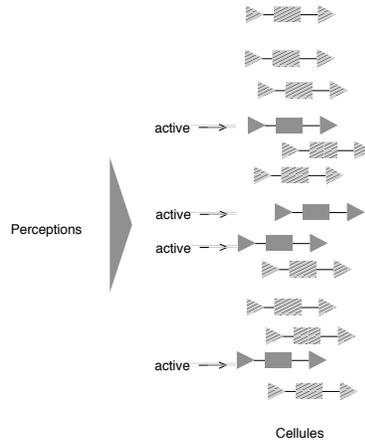


FIG. 9.6: A tout moment les données sensorielles déterminent un ensemble de cellules actives,  $\mathcal{S}_B(X)$ .

$$\mathcal{S}_B(X) = \{cell_i \mid cell_i \in \mathcal{C}_B \text{ et } active(cell_i)\} \quad (9.6)$$

## Histoire

Une cellule  $cell_i$  n'est active que dans certaines frames  $\{Y, X\}$  de l'ensemble de démonstration  $D_B$ . En énumérant ces frames, on peut former la liste des actions correspondant aux frames où  $cell_i$  est active. Cette liste appelée *histoire de la cellule*  $cell_i$  pour le comportement  $B$  et notée  $\mathcal{H}_B(cell_i)$ .

$$\mathcal{H}_B(cell_i) = \{Y \mid \{Y, X\} \in D_B \wedge cell_i \in \mathcal{S}_B(X)\} \quad (9.7)$$

### 9.2.3 Mécanisme de synthèse

Le mécanisme de synthèse construit la population de cellules à partir des démonstrations. Il consiste d'une part à déterminer les cellules utiles pour le comportement visé puis à associer à chaque cellule l'action appropriée. Le nombre de micro-percepts possibles et par conséquent le nombre de cellules est très grand, ce qui ne rend pas envisageable de les considérer toutes. Le modèle doit donc fournir un mécanisme de sélection initial des cellules utiles à partir des données sensorielles contenues dans les démonstrations. L'ensemble  $\mathcal{C}_B$  est d'abord construit à partir des démonstrations de  $B$  par l'algorithme *generate\_pop* (eq 9.8) chargé de la sélection. Une fois cette population de cellules générée, chaque cellule est alors informée avec l'action

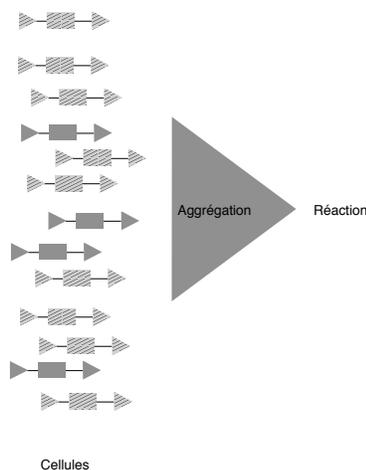


FIG. 9.7: Lors de la réalisation autonome du comportement, les actions des cellules actives sont agrégées pour déterminer l'action courante.

qui lui correspond par l'algorithme *inform\_pop*.

$$\mathcal{C}_B \leftarrow \text{generate\_pop}(D_B) \quad (9.8)$$

$$\mathcal{C}_B \leftarrow \text{inform\_pop}(D_B, \mathcal{C}_B) \quad (9.9)$$

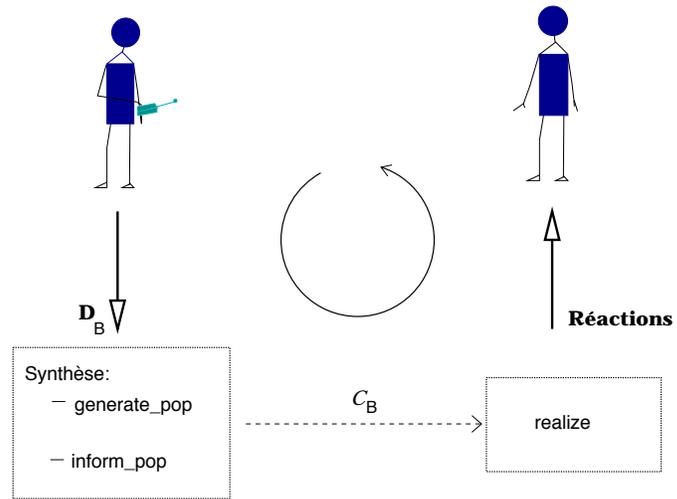
#### 9.2.4 Mécanisme de réalisation autonome

Lorsque le robot reproduit le comportement appris, il est dans une *phase de réalisation autonome*. Celle-ci consiste à déterminer l'action à effectuer en fonction des perceptions, c'est à dire à déterminer une valeur pour  $Y_t$  en fonction des données sensorielles courantes et éventuellement passées. Cette réaction est obtenue en *agrégeant* les actions proposées par les cellules activées (fig. 10.5). Elle est calculée par l'algorithme *realize* (eq. 9.10). L'action courante est donc déterminée par la contribution collective des cellules actives.

$$Y_t \leftarrow \text{realize}(\mathcal{S}_B) \quad (9.10)$$

#### 9.2.5 Schéma général

Les algorithmes *generatepop*, *inform\_pop* et *realize* s'intègrent dans le processus de synthèse supervisée selon le schéma général de la figure 9.8. Ils seront détaillés dans deux chapitres

FIG. 9.8: Schéma général de la synthèse d'un comportement *B*

suivants.



# Chapitre 10

## MPL-t Modèle taxique

### 10.1 Introduction

Cette première variante du modèle est appelé taxique par analogie avec le terme utilisé en éthologie. Une taxie ou un tropisme est une réaction de locomotion orientée par un stimulus extérieur. Il a été relevé chez les animaux et les plantes de nombreux comportements taxiques, tels que le comportement phototaxique d'un papillon attiré par la lumière. Tôt dans le siècle, des éthologues, tels que Jacques Loeb, ont proposé que tous les comportements pouvaient résulter de la seule combinaison de plusieurs tropismes élémentaires, cette généralisation à été cependant contredite depuis (voir [Ruwet, 1975]).

Dans le modèle **MPL-t** cette idée est transposée en considérant que les points du champ visuel du robot sont autant de stimuli qui influencent indépendamment les actions du robot. Le principe du modèle est le suivant : lorsque le robot réalise le comportement, une action est associée à chaque point de l'image perçue, si ce point correspond à une perception connue.

L'ensemble des points du champ perceptuel détermine de cette manière l'équivalent d'une affordance indiquant à partir des perceptions courantes une action possible. Dans l'exemple présenté dans les figures suivantes (10.1 à 10.4), exemple issu d'expérimentations réelles, le robot a appris à passer par une porte à partir de quelques démonstrations. Lors de la réalisation autonome, certaines cellules sont actives selon les situations perceptuelles : lorsque le robot est éloigné de la porte, les cellules actives "proposent" d'avancer, alors que d'autres "proposeront" de tourner fortement quand le robot est près de la porte.



FIG. 10.1: Le robot à appris un comportement de passage de porte à partir de quelques démonstrations. La figure montre ce que 'voit' le robot lorsque celui-ci est en phase de réalisation autonome et situé loin de la porte.

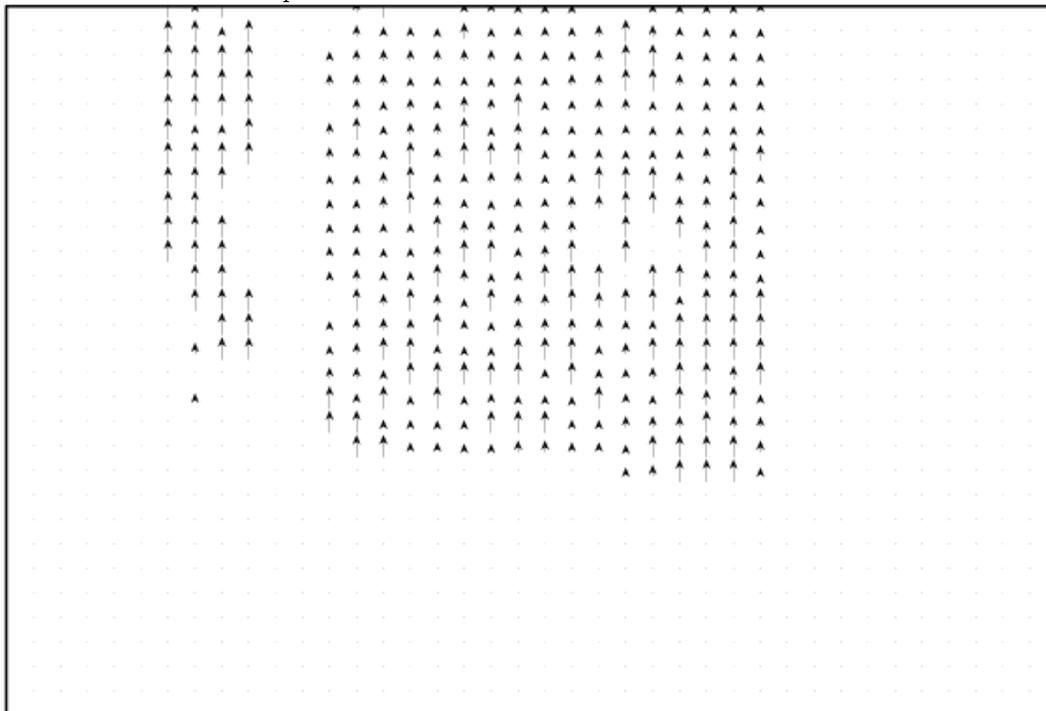


FIG. 10.2: Réactions correspondantes proposées par les cellules. La figure visualise pour chaque cellule active le vecteur action qu'elle a mémorisé. Les cellules sont directement activées par les caractéristiques de couleur du point correspondant de l'image à la même position. Les cellules ayant un gain trop faible ne sont pas visualisée (d'où les zones vides).

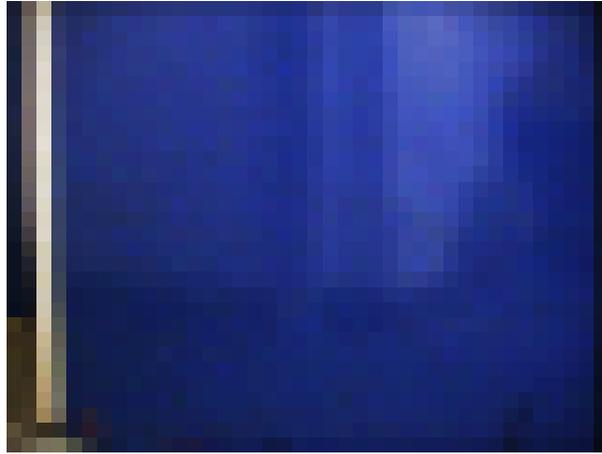


FIG. 10.3: Perception visuelle du robot lorsque celui-ci s'apprête à passer la porte

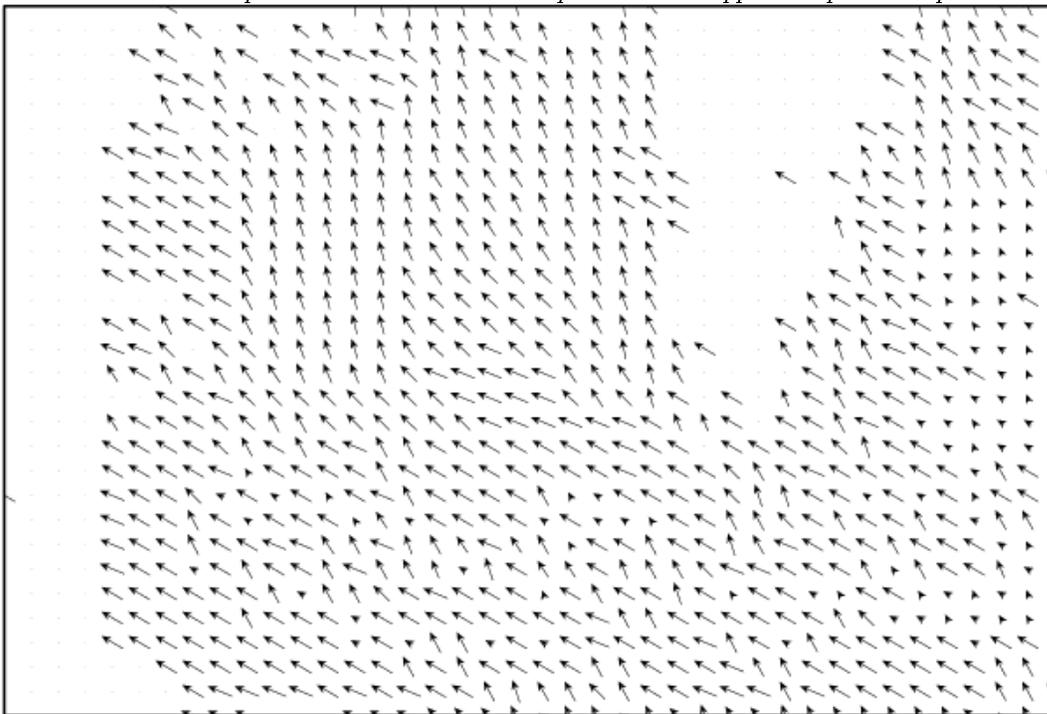


FIG. 10.4: Réaction proposée par les cellules

Chaque cellule ne correspond qu'à un point unique du champ perceptuel. Par conséquent, elle ne fait référence qu'à un seul micro-percept (eq. 10.1).

$$cell_i \equiv (mp_1^i, y_i, w_i) \quad (10.1)$$

Si le modèle est mis en oeuvre avec le champ perceptuel CP1, le micro-percept contenu dans chaque cellule est alors formé par la conjonction de deux tessels ayant la même position :

$$mp_i \equiv (Couleur(pos) = valeur\_couleur) \wedge (Forme(pos) = valeur\_forme) \quad (10.2)$$

## 10.2 Mécanisme de synthèse

Le mécanisme de synthèse du comportement  $B$  à partir des démonstrations consiste d'abord à construire la population de cellules en fonction des micro-percepts présents dans les démonstrations puis à déterminer la valeur de  $y_i$  et  $w_i$  en fonction de ces mêmes démonstrations.

### 10.2.1 Selection des cellules : *generate\_pop*

Pour construire la population de cellules  $\mathcal{C}_B$  on forme l'ensemble de tous les *micro-percepts distincts apparaissant au moins une fois dans les démonstrations*. Pour chacun des micro-percepts de cet ensemble on crée ensuite la cellule correspondante. Ainsi, si le micro-percept  $mp \equiv (Couleur(25, 30) = bleu) \wedge (Forme(25, 30) = 12)$  apparaît au moins une fois dans l'une des démonstrations, il donnera lieu à la création d'une (et une seule) cellule. On notera que le nombre de micro-percepts apparaissant effectivement dans les démonstrations est beaucoup moins grand que le nombre de micro-percepts potentiels.

### 10.2.2 Détermination de l'action : *inform\_pop*

Le mécanisme d'apprentissage consiste ensuite à établir la valeur de  $y_i$  pour chacune des cellules de  $\mathcal{C}_B$ . Chaque cellule mémorise *l'action moyenne* effectuée dans les démonstrations lorsque la cellule est active. La cellule n'est active qu'à certains moments et l'action moyenne est calculée sur ces seules périodes d'activité (zones grisées de la figure 10.5).

Plus formellement, on peut considérer  $y_i$  comme correspondant à l'espérance de l'effecteur

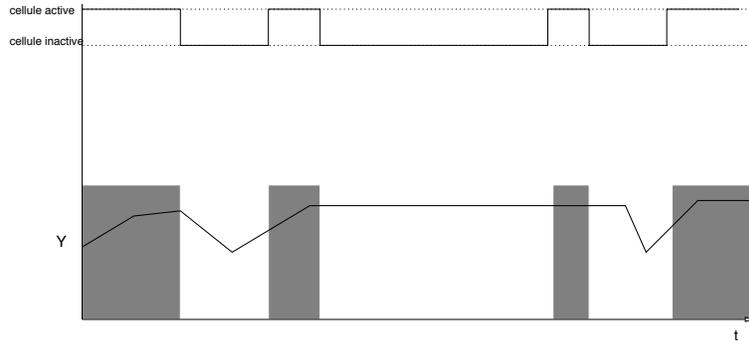


FIG. 10.5: périodes d'activité et d'inactivité d'une cellule. L'action moyenne est calculé sur les périodes grisées.

$Y$  conditionnée par l'activation de la cellule  $cell_i$  (eq. 10.3).

$$y_i = E[Y|active(cell_i)] \quad (10.3)$$

La distribution de probabilité  $F(Y|active(cell_i))$  est inconnue et on doit estimer l'espérance à partir des données empiriques des démonstrations. Cette information est obtenue à partir de l'histoire  $\mathcal{H}_B(cell_i)$  de chaque cellule, en calculant sur cet ensemble d'actions la moyenne empirique de l'effecteur  $Y$ . L'algorithme *inform-pop* consiste donc à calculer cette valeur pour chaque cellule, comme indiqué dans l'équation 10.4.

$$y_i = \frac{1}{|\mathcal{H}_B(cell_i)|} \sum_{\mathcal{H}_B(cell_i)} Y \quad (10.4)$$

### 10.2.3 Détermination du gain

Plusieurs stratégies sont envisageables pour déterminer le gain  $w_i$  de chaque cellule. Il peut tout simplement être fixé à 1 pour prendre en compte de manière équivalente chaque cellule. Il peut servir à favoriser les cellules les plus fréquemment utilisées dans les démonstration s'il correspond au nombre d'activations de  $cell_i$  sur  $D_B$ . Il peut permettre d'associer aux cellules une informations de *saillance*, c'est-à-dire de moindre ou plus grande importance d'une perception par rapport aux autres pour une action donnée.

Pour l'exemple présenté dans les figures 10.1 à 10.4 le gain utilisé correspond à un critère empirique de saillance portant uniquement sur le tessel couleur mémorisé dans le micro-

percept. Le gain est d'autant plus fort que la saturation<sup>23</sup> de la couleur est importante et que la couleur apparaît dans des zones compactes<sup>24</sup>. Autrement dit, il favorise les zones compactes de couleurs saturées.

### 10.3 Mécanisme de réalisation autonome

Lors de la réalisation autonome, le comportement doit produire une *réaction* correspondant aux données sensorielles courantes  $X$ . Cette réaction, notée  $\hat{Y}$ , est obtenue par agrégation, en faisant la moyenne des actions proposées par les cellules actives, celles appartenant au support  $\mathcal{S}_B(X)$ . La moyenne est pondérée pour chaque cellule par son gain. L'algorithme *realize* consiste à calculer  $\hat{Y}$  comme indiqué dans l'équation 10.5.

$$\hat{Y} = \frac{1}{Norm} \sum_{\mathcal{S}_B(X)} y_i \times w_i \quad (10.5)$$

où *Norm* sert de terme normalisateur.

$$Norm = \sum_{\mathcal{S}_B(X)} w_i \quad (10.6)$$

### 10.4 Application à un exemple élémentaire

Un exemple très simplifié permet d'illustrer le fonctionnement du modèle. On considère un champ perceptuel minimal réduit à un seul plan perceptuel *Couleur* de dimension  $2 \times 2$ . Le micro-percept de chaque cellule est donc de la forme  $Couleur(pos) = couleur$ . Pour simplifier, le gain  $w_i$  est fixé à 1. Le comportement à apprendre consiste à s'approcher d'un cube puis à ralentir. Pour cela on dispose d'une seule démonstration contenant 4 frames successives et représentées dans la figure 10.6.

---

<sup>23</sup> Au sens du codage HSV (hue, saturation, value).

<sup>24</sup> rapport du nombre de pixels d'une couleur sur la surface totale de l'image.

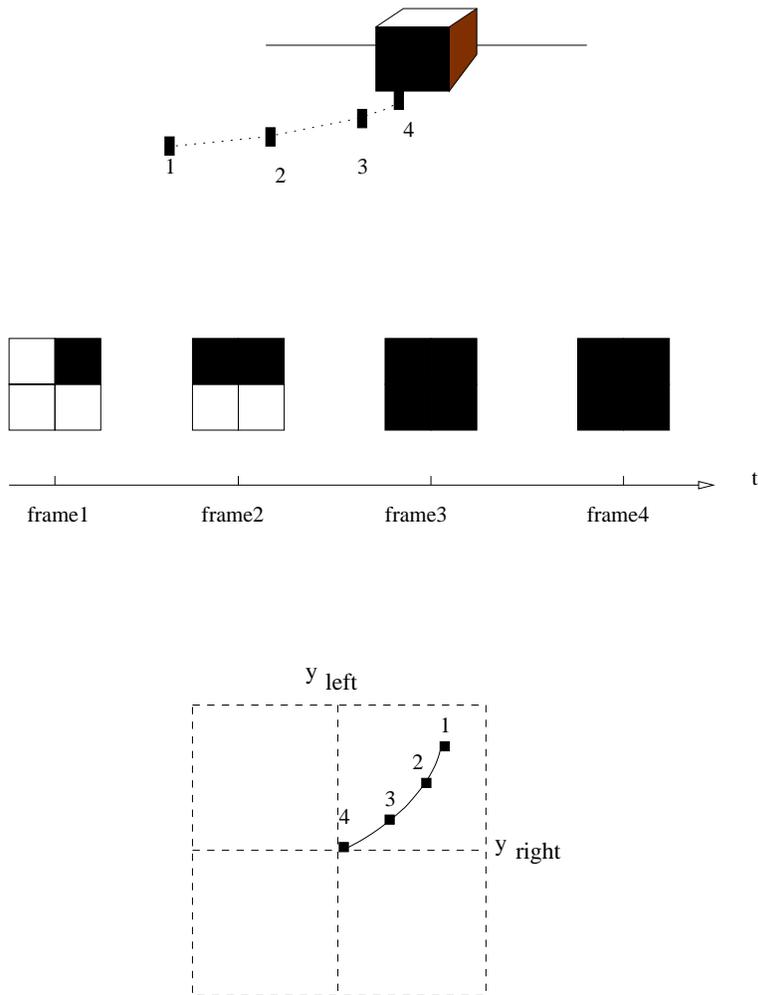


FIG. 10.6: Les quatre frames constituant la démonstration : (en haut) la trajectoire de la démonstration et les quatre points correspondant aux frames ; (au milieu) l'image  $X$  perçue pour chaque frame ; (en bas) la valeur des effecteurs pour les quatre frames.

frame	$Y_{left}$	$Y_{right}$	correspond à :
1	0.7	0.7	avance
2	0.5	0.6	tourne un peu à gauche
3	0.2	0.3	tourne à gauche lentement
4	0.0	0.0	arrêt

TAB. 10.1: Les ordres moteurs produits par le tuteur pour les quatre frames de la démonstration.

$cell_i$	$mp_i$	apparaît dans frame	actif dans frames
$cell_1$	$Couleur(0, 0) = blanc$	1	1
$cell_2$	$Couleur(0, 1) = noir$	1	1,2,3,4
$cell_3$	$Couleur(1, 0) = blanc$	1	1,2
$cell_4$	$Couleur(1, 1) = blanc$	1	1,2
$cell_5$	$Couleur(0, 0) = noir$	2	2,3,4
$cell_6$	$Couleur(0, 1) = noir$	3	3,4
$cell_7$	$Couleur(1, 1) = blanc$	3	3,4

 TAB. 10.2: Les sept cellules créées par `generate_pop` et leur micro-percept.

Les actions correspondant issues du contrôle du tuteur sont récapitulées dans la table 10.1 sous la forme de la vitesse de chaque roue. Dans la phase d'apprentissage, l'algorithme `generate_pop` donne lieu à la création de 7 cellules (table 10.2) dont la partie action est ensuite informée par `inform_pop` conformément à la table 10.3.

Lors de la réalisation autonome du comportement, le robot peut se trouver dans une situation perceptuelle telle que celle de la figure 10.7, cas où le robot est proche du cube. Le support perceptuel est alors constitué des 4 cellules actives (eq. 10.7) et la réaction du robot sera calculée à partir de ces quatre cellules conformément aux équations 10.8 et 10.9. Le mouvement résultant  $Y_{left} = 0.31$  et  $Y_{right} = 0.35$  correspond à une approche très lente du cube.

$cell_i$	$y_{left}$	$y_{right}$	$gain_i$
$cell_1$	0.7	0.7	1
$cell_2$	0.35	0.4	1
$cell_3$	0.6	0.65	1
$cell_4$	0.6	0.65	1
$cell_5$	0.23	0.3	1
$cell_6$	0.1	0.15	1
$cell_7$	0.1	0.15	1

 TAB. 10.3: Les sept cellules informées par `inform_pop`.

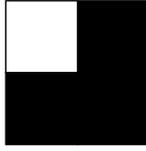


FIG. 10.7: Perception rencontrée par le robot lors de la phase de réalisation autonome.

$$\mathcal{S}_B(X) = \{cell_1, cell_2, cell_6, cell_7\} \quad (10.7)$$

$$Y_{left} = \frac{0.7 + 0.35 + 0.1 + 0.1}{4} = 0.31 \quad (10.8)$$

$$Y_{right} = \frac{0.7 + 0.4 + 0.15 + 0.15}{4} = 0.35 \quad (10.9)$$

## 10.5 Complexité

Dans le cadre de la synthèse supervisée de comportements, il est très important que la complexité en temps de la phase d'apprentissage soit la plus faible possible. C'est surtout la complexité de l'apprentissage en fonction du nombre de frames  $m$  constituant les démonstrations qui importe, car elle peut varier de manière significative, traduisant ainsi la complexité du comportement visé (plus long ou avec plus de variantes). Il suffit d'une passe sur les frames de  $D_B$  pour créer les cellules nécessaires et d'une autre pour calculer la moyenne  $y_i$  pour chaque cellule<sup>25</sup>. Par conséquent, les algorithmes *generate\_pop* et *inform\_pop* ont tous deux une complexité linéaire en fonction de  $m$ .

La méthode d'accès aux cellules est, elle aussi, très importante. En effet, dans la phase d'apprentissage ainsi que dans la phase de réalisation, il est nécessaire de déterminer les cellules actives (ie :  $\mathcal{S}_B(X)$ ) d'après les micro-percepts effectivement présents dans les perceptions courantes. Les fonctions d'accès aux cellules de  $\mathcal{C}_B$  réalisables ont une complexité en temps qui dépend du nombre de cellules  $n = |\mathcal{C}_B|$ . En utilisant une fonction d'accès par hachage, les complexités en temps de *generate\_pop* et *inform\_pop* en fonction de  $m$  et  $n$  sont en moyenne en  $O(mn/h)$  (où  $h$  est le nombre de clés de hachage).

<sup>25</sup>Ces deux passes pouvant être aisément combinées en une seule passe.

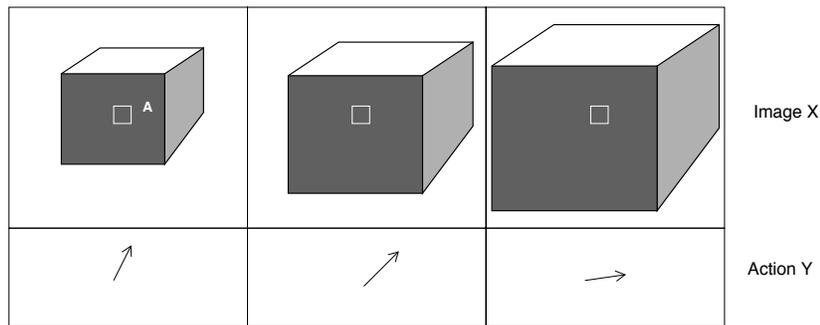


FIG. 10.8: Une démonstration difficile à traiter par cette variante ( $X$  en haut,  $Y$  en bas). Les caractéristiques du point  $A$  sont identiques dans des situations très différentes.

## 10.6 Remarques

La variante **MPL-t** repose sur l'association d'une action à un point du champ perceptuel. Elle traduit concrètement l'intuition selon laquelle les éléments de l'environnement influencent directement les actions du robot. Comme on le voit dans les figures du paragraphe 10.1, le comportement est interprétable sur la base des informations produites par chaque cellule. Bien qu'il n'y ait pas de reconnaissance d'objets proprement dite, un groupe de tessels agira un peu comme un objet puisque dans les faits le robot choisira son action en fonction de la position de ce groupe.

Cependant, plusieurs problèmes se posent. Le nombre de micro-percepts à traiter, même s'il est réduit par la redondance des tessels (un même tessel apparaîtra plusieurs fois dans les démonstrations) devient assez vite grand et engendre une population de cellules importante. Par ailleurs, dans cette variante, les points du champ perceptuel sont considérés *indépendamment* les uns des autres. On retrouve ici une similarité avec l'apprentissage Bayésien naïf ([Mitchell, 1997] pp 180) utilisé dans des applications de classification de texte où l'occurrence d'un mot à l'une des  $n$  positions du texte est considéré indépendamment des mots trouvés aux autres positions. Avec ces micro-percepts, aucune relation n'est établie entre les points et ceci peut produire des comportements incorrects. Ainsi, dans une démonstration (fig. 10.8) où le robot doit s'approcher d'un objet puis tourner, le point marqué  $A$  conserve les mêmes caractéristiques locales (couleur, environnement, etc.) dans des situations différentes. De ce fait la moyenne des actions qui sera utilisée dans une situation comme celle de la figure 10.9 ne sera pas celle attendue.

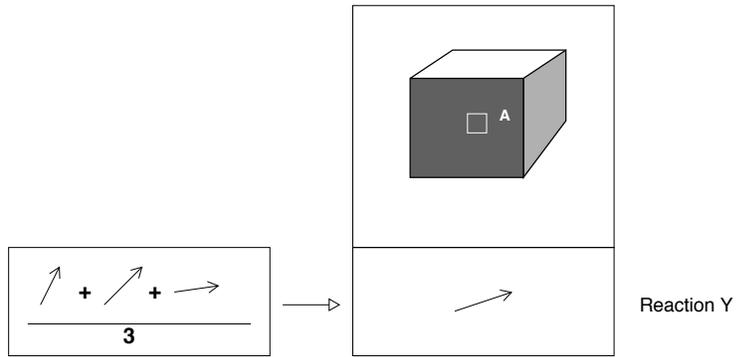


FIG. 10.9: *Reaction de la cellule correspondant au point A est inappropriée.*

On peut palier partiellement ces effets en agissant sur le type de plans perceptuels utilisés ou encore sur la nature du gain utilisé, mais cette amélioration reste trop dépendante de l'environnement considéré. Le chapitre suivant présentera avec la variante **MPL-s** une extension plus générique et plus robuste.



# Chapitre 11

## MPL-s Modèle à échantillons

### 11.1 Introduction

La variante **MPL-s** est proposée pour répondre aux limitations évoquées dans la section 10.6. Celles-ci sont principalement dûes au fait que les points du champ perceptuel sont considérés indépendamment les uns des autres : une même cellule peut être activée dans des contextes trop différents. Le modèle **MPL-t** est étendu pour prendre en compte ce problème ainsi que pour diminuer le nombre de cellule générées durant la phase d'apprentissage. Cette extension est réalisée en associant plusieurs points du champ perceptuel à une même cellule et en substituant une stratégie d'échantillonnage à l'énumération exhaustive des micro-percepts.

### 11.2 Sélectivité de percepts à n pixels pris aléatoirement

Il est possible de distinguer efficacement plusieurs perceptions visuelles complexes en utilisant des percepts très simples constitués de 2, 3 ou 4 pixels choisis aléatoirement dans une image. Supposons, par exemple, que le robot perçoive une image comme celle la figure 11.1 et qu'il note la valeur et la position de deux pixels  $p_1$ ,  $p_2$ . Si par la suite il se déplace dans son environnement, il ne retrouvera ces deux mêmes pixels que dans très peu de situations. Le fait d'associer deux pixels entre eux impose une contrainte très forte et restreint le nombre d'images qui y correspondent. Ce fait peut être mesuré expérimentalement grâce au test *Select* décrit ci-après.

Le test *Select* décrit dans l'encadré 1 mesure la capacité d'un percept constitué de  $p$  pixels

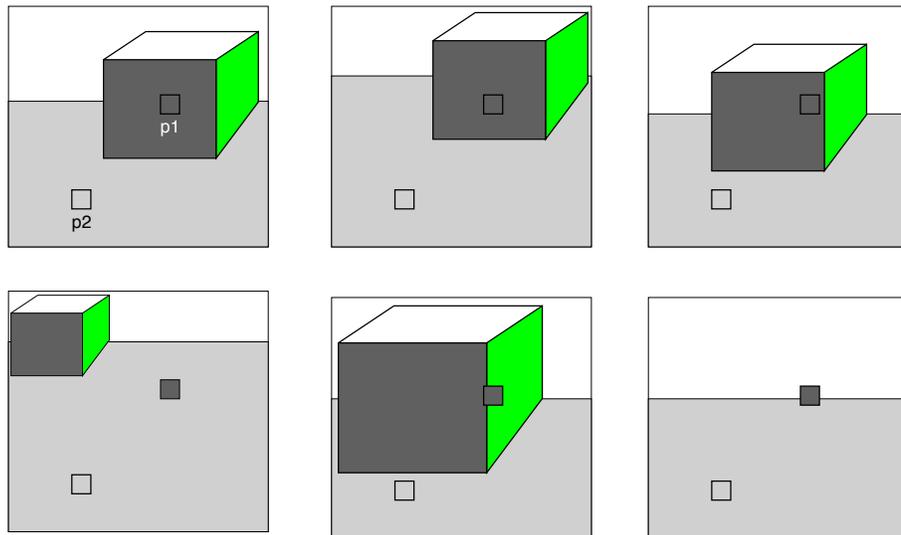


FIG. 11.1: Le percept formés des deux pixel  $p1$  et  $p2$  est retrouvé à l'identique dans les trois images du haut, très similaires du point de vue du robot, mais pas dans celles du bas.

valués choisis aléatoirement à discriminer une image dans un ensemble d'images. Un pixel valué est ici un t-uple  $(x, y, couleur)$ . Le test est effectué sur un ensemble  $S$  de  $n = 1200$  images semblables à celles de la figure 11.2 et obtenues au cours de sessions de marche aléatoire du robot dans différents points du laboratoire. Ces images sont caractéristiques du point de vue du robot dans un environnement structuré.

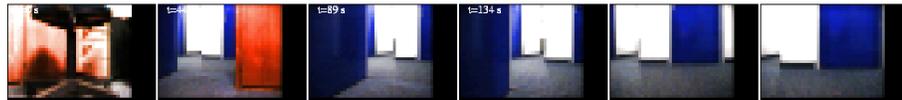


FIG. 11.2: Quelques images de l'ensemble  $S$  provenant de la caméra du robot .

- 1: Choisir aléatoirement une image dans l'ensemble  $S$ .
- 2: Dans cette image choisir aléatoirement et mémoriser  $p$  pixels valués.
- 3: Sur l'ensemble  $S$ , compter le nombre  $r$  d'images contenant les  $p$  pixels, (même couleur, même position).
- 4: Retourner le pourcentage d'identification d'une image de  $S$  :  $\left[1 - \frac{r}{n}\right] \times 100$  .

**Algorithme 1:**  $Select(p, S)$

La fonction  $Select(p, S)$  est reproduite dans la figure 11.3. Pour chaque point de la courbe

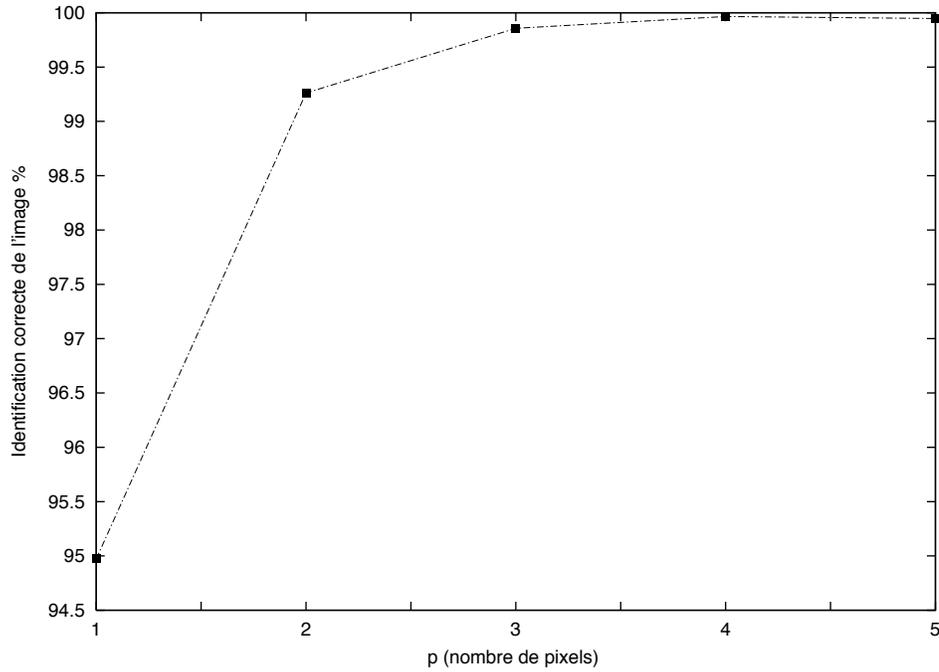


FIG. 11.3: Estimation de  $E[\text{Select} \mid p, S]$  sur un ensemble de 1200 images collectées par le robot mobile.

les résultats sont moyennés sur 150 essais pour une meilleure validité statistique. Le tracé montre la valeur moyenne pour  $p = 1, 2, 3, 4$  et 5. On constate donc qu'*en moyenne*, 2 pixels sont suffisants pour discriminer de manière satisfaisante une image parmi  $n$ .

### 11.3 Nouvelle structure de la cellule

La nouvelle structure découle directement du constat du précédent paragraphe. Elle reprend l'idée de percepts formés de quelques pixels en la transposant dans la cellule. La conjonction de micro-percepts  $x_i$  est maintenant formée de  $p$  micro-percepts, chacun correspondant à une position différente (éq. 11.2). Cette modification mineure en apparence change profondément les conditions dans lesquelles sont activées les cellules.

$$cell_i \equiv (x_i, y_i, w_i) \quad (11.1)$$

$$x_i = mp_1^i(pos_1) \wedge mp_2^i(pos_2) \wedge \cdots \wedge mp_p^i(pos_p) \quad (11.2)$$

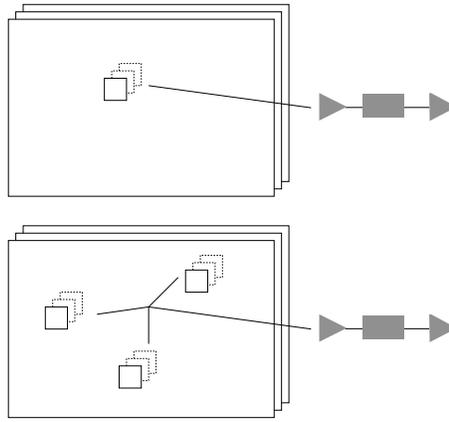


FIG. 11.4: On passe d'une cellule comportant un micro-percept (en haut) à une cellule qui en comporte plusieurs (en bas).

## 11.4 Mécanisme de synthèse

La principale conséquence du changement de structure porte sur le mécanisme de sélections des cellules. En effet il n'est plus possible d'examiner les occurrences de tous les micro-percepts survenant dans les démonstrations pour déterminer les cellules à créer. Si le champ perceptuel est rectangulaire et comporte  $l \times h$  positions discrètes, on peut alors y détecter  $\binom{p}{l \times h}$  micro-percepts distincts.

La stratégie suivie consiste à *échantillonner* les frames provenant des démonstrations. L'algorithme *generate\_pop* de génération de la population de cellules est directement inspiré du test *Select*. Comme décrit dans l'encadré 2, la partie  $x_i$  de chaque cellule est construite en choisissant aléatoirement une des positions dans une frame elle-même choisie au hasard.

- 1: Estimer le nombre de cellules à créer  $m$ .
- 2: **pour** chacune des  $m$  cellules  $cell_i$
- 3:   **répéter**
- 4:     Choisir une frame aléatoirement dans  $D_B$ .
- 5:     Choisir  $p$  positions aléatoirement dans cette frame.
- 6:     Relever les  $p$  micro-percepts correspondants et former  $x_i$ .
- 7:     **jusqu'à ce que** la contrainte  $C(x_i)$  soit satisfaite.
- 8:     Créer la cellule correspondante.
- 9: **finpour**

**Algorithme 2:** *generate\_pop<sub>2</sub>*

La détermination de l'action demeure inchangée, elle repose toujours sur l'algorithme

*inform\_pop* et consiste à calculer la moyenne des actions pour les période d'activité de la cellule.

### Contrainte $C(x_i)$

La contrainte  $C$  porte sur la conjonction de micro-percepts  $x_i$ . Elle peut être ignorée mais on peut l'utiliser pour structurer la conjonction de micro-percepts en imposant des conditions sur chacun d'eux. Ces conditions peuvent porter sur la distance euclidienne entre les micro-percepts comme dans la contrainte de l'équation 11.3.

$$C_1(x_i) \equiv \forall (mp_j^i, mp_k^i) \in x_i, d_{min} < distance(mp_j^i, mp_k^i) < d_{max} \quad (11.3)$$

Puisque l'échantillonnage est uniforme, la distribution des tessels utilisés dans la population de cellules est similaire à la distribution des tessels présents dans les démonstrations. Il est intéressant de biaiser cet échantillonnage pour défavoriser des zones qui seraient sur-représentées dans les démonstrations et porteuses de peu d'information, telles que les murs ou le sol. Ceci peut être intégré dans une contrainte, à partir de l'histogramme des tessels pour chaque plan perceptuel. Ainsi, si on veut que toutes les couleurs présentes soient à peu près également représentées on vérifiera pour chaque micro-percept que la valeur correspondante de l'histogramme du plan *Couleur* ne dépasse pas un seuil donné (eq. 11.4).

$$C_2(x_i) \equiv \forall mp_j^i \in x_i, Histo_{Couleur}(mp_j^i) < \frac{1}{nbcouleurs}. \quad (11.4)$$

Finalement les différentes contraintes peuvent être combinées (eq. 11.5).

$$C(x_i) = C_1(x_i) \wedge C_2(x_i) \quad (11.5)$$

### Estimation du nombre de cellule

Le nombre de cellules,  $m = |\mathcal{C}_B|$ , doit être estimé à partir des démonstrations. Dans la variante **MPL-t** ce nombre pouvait poser un problème puisqu'il correspondait au nombre  $|MP|$  de micro-percepts différents relevés dans les démonstrations, celui-ci pouvant être relativement important. La stratégie d'échantillonnage retenue ici permet de s'affranchir de cette difficulté en déterminant en partie arbitrairement  $m$ . Ainsi, en gardant comme indicateur de complexité

du comportement le nombre de micro-percepts que l'on multiplie par une constante arbitraire  $\alpha$  (empiriquement j'ai utilisé la valeur  $\alpha = 0.20$ ),  $m$  peut être déterminé par :

$$m = \alpha |MP| \quad (11.6)$$

## 11.5 Mécanisme de réalisation autonome

Le mécanisme de réalisation autonome suit exactement le même principe que celui de la variante **MPL-t**. La réaction du robot correspondra ici aussi à la moyenne pondérée des actions proposées par les cellules actives.

Le nombre de cellules est suffisamment important pour que plusieurs cellules soient détectées dans une situation donnée - comme c'est le cas dans la figure 11.5. Les cellules détectées identifient précisément l'image car peu de postures pourraient donner lieu à une perception identique. En présence de bruit certaines cellules ne seront pas activées et le support  $\mathcal{S}_B$  sera seulement plus petit.

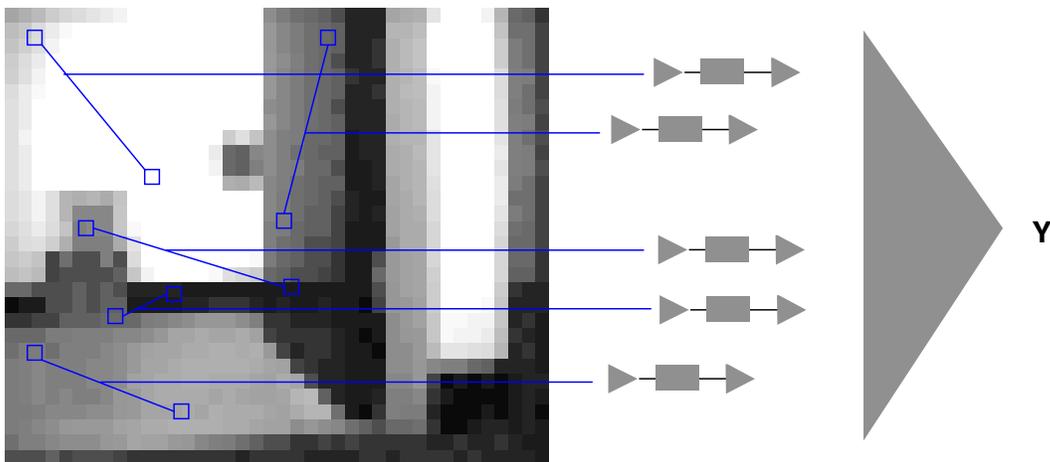


FIG. 11.5: Réalisation autonome : les cellules activées par les données sensorielles déterminent collectivement l'action courante.

## 11.6 Remarques

Pour paramétrer le modèle **MPL-s** il faut définir  $p$  le nombre de micro-percept par cellules,  $\alpha$  le taux d'échantillonnage et définir éventuellement les paramètres de la contraintes  $C(x_i)$ . Par ailleurs il est clair que la façon dont on structure le champ perceptuel est cruciale et conditionne le fonctionnement du modèle.

## Chapitre 12

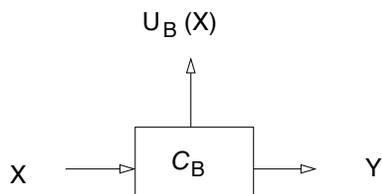
# Compléments au modèle d'acquisition

### 12.1 Introduction

Ce chapitre introduit plusieurs compléments au modèle **MPL** proposé dans les précédents chapitres. Ces compléments reposent principalement sur l'utilisation d'une *mesure d'utilité*. Ils ont pour fonction, d'une part, de fournir un meilleur contrôle de la réalisation autonome en introduisant la possibilité d'une perception active, d'autre part, de faciliter l'intégration des comportements en tant que composants d'une architecture plus large.

### 12.2 Mesure d'utilité

Pour contrôler le robot, il est important de disposer d'une information indiquant le degré de correspondance entre le comportement en cours et celui envisagé par le tuteur au travers des démonstrations. Cette correspondance peut être exprimée par une information indiquant le degré d'utilité des perceptions courantes pour un comportement  $B$  donné. Dans la plupart des modèles purement numériques, ce type d'information n'est pas directement disponible car on ne sait pas ce qui est important dans les données sensorielles courantes, qui sont toutes traitées de la même manière. Au contraire, le modèle **MPL** fournit une information permettant de mesurer la pertinence des données sensorielles courantes. Cette information est déterminée d'après les cellules actives : la mesure d'utilité  $U_B(X)$  est définie comme étant le nombre de cellules de  $B$  activée par les données sensorielles courantes  $X$ . Elle correspond donc au

FIG. 12.1: Mesure  $U_B(X)$  obtenue en mode autonome

cardinal du support (eq. 12.1). La mesure  $U_B(X)$  sera faible si le robot se trouve devant une scène totalement nouvelle puisque peu ou pas de cellules seront actives. A l'inverse  $U_B(X)$  sera élevée si la scène appartient à l'une des démonstrations. Quand le robot percevra une image partielle ou décalée par rapport aux exemples,  $U_B(X)$  aura une valeur moyenne.

$$U_B(X) = |\mathcal{S}_B(X)| \quad (12.1)$$

### 12.3 Perception active

Au cours de la réalisation autonome d'un comportement, le robot peut se trouver dans des situations où ses perceptions sont difficiles à exploiter. Cela peut se produire si par exemple quelqu'un passe près du robot ou encore s'il se trouve en face d'une scène inconnue. Ce type de situations est nécessairement assez fréquent puisque, d'une part, le robot n'a connaissance au travers des démonstrations que de quelques situations types et que d'autre part la dynamique de l'environnement implique qu'il se trouve souvent confronté à des situations au moins partiellement inconnues. La dynamique propre du robot est elle-même imprévisible, un glissement ou un à-coup pouvant changer radicalement sa vision. Dans ces cas, le robot a besoin d'un moyen pour revenir dans une trajectoire plus exploitable et il peut alors utiliser la mesure d'utilité  $U_B(X)$  pour décider d'une action à entreprendre. Si  $U_B(X)$  est en dessous d'un seuil donné, le robot peut (en se déplaçant sur lui-même), chercher une meilleure position avant de continuer à progresser. Ce type de perception est souvent appelée *perception active* en robotique en opposition au traitement passif des entrées [Ballard and Brown, 1993].

Un exemple d'application de ce principe est donné ci-dessous. La perception active est réalisée en pivotant la caméra (il serait aussi possible mais plus long de faire pivoter le robot sur lui-même). Un comportement  $B$  a été appris dans l'environnement de la figure 12.2 avec le modèle **MPL-s**. Le comportement consistait à rentrer dans la pièce A et à s'approcher

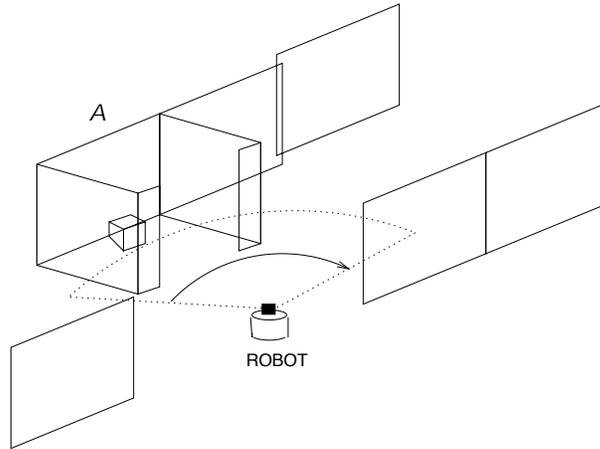


FIG. 12.2: Mouvement panoramique de la caméra du robot effectué après l'apprentissage d'un comportement  $B$  devant l'environnement d'apprentissage constitué par une petite pièce dans un couloir.

de la boîte qui s'y trouve. Lors de la réalisation autonome, le robot obtient les perceptions successives de la figure 12.3 en effectuant un mouvement panoramique dans cet environnement. Pour chacune de ces vues un nombre de micro-percepts (points clairs dans les images) plus ou moins grand est détecté. Ceci est reflété dans le tracé de la figure 12.4 correspondant à la mesure  $U_B(X)$  pour ces mêmes vues.  $U_B(X)$  est maximale pour les vues qui correspondent le mieux aux exemples et qui peuvent être utilisées par le comportement.

En dérivant ce principe, il est possible d'étendre l'algorithme de réalisation autonome *Realize*. Comme indiqué dans l'automate de la figure 12.5, le robot se trouve par défaut dans l'état *Realize*, mais si  $U_B(X)$  passe en-dessous d'un seuil donné il passe dans l'état *Pan*. Il se met alors à rechercher autour de lui en bougeant la caméra pour couvrir petit à petit son champ visuel en appliquant un comportement semblable à celui de la figure 12.6. Dès qu'il trouve une situation où  $U(X)$  atteint un maximum local, il tourne sur lui-même pour se repositionner.

## 12.4 Sélection de comportements

Un système robotique capable de réaliser plusieurs comportements doit être capable d'activer les comportements appropriés en fonction des circonstances et du contexte. Ce problème est similaire au problème général de *sélection de l'action* en robotique située [Maes, 1994].

La mesure  $U_B(X)$  peut être encore utilisée pour faciliter la sélection du comportement



FIG. 12.3: Vues successives obtenues pendant le mouvement panoramique. Les point clairs indiquent les positions des micro-percepts détecté.

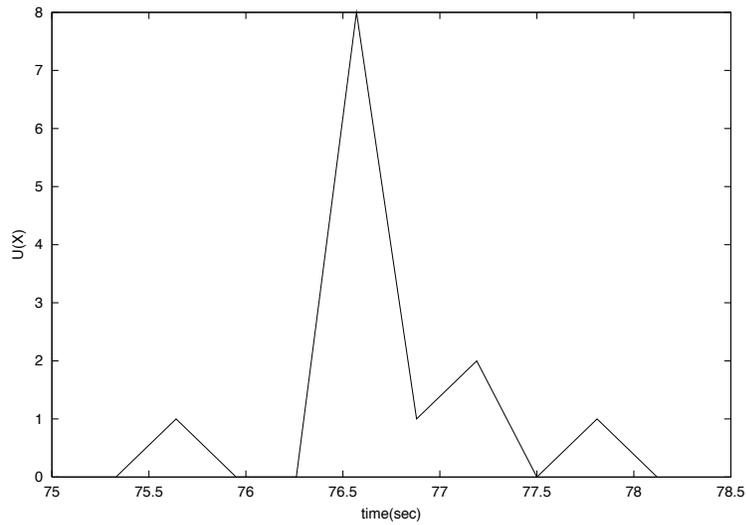


FIG. 12.4: Mesure de  $U_B(X)$  pendant le mouvement panoramique, elle est maximum dans la position qui correspond le mieux au comportement.

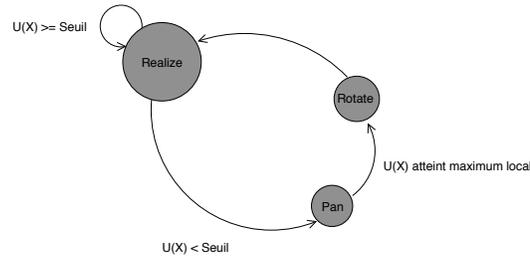


FIG. 12.5: Diagramme d'états de la réalisation autonome avec perception active.

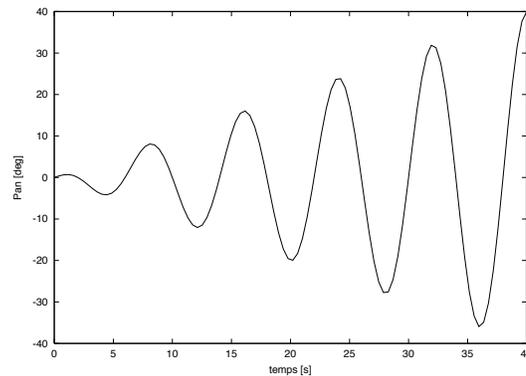


FIG. 12.6: Angle  $a$  de la camera en degrés en fonction du temps dans l'état Pan -  $a = \cos(\pi t/4)t$ .

approprié. La figure 12.4 montre la fonction  $U_B(X)$  mesurée en situation réelle pour deux comportements  $B1$  et  $B2$  correspondant à deux environnements distincts. La mesure est effectuée dans l'environnement de  $B1$  pour les deux comportements et indique que  $B1$  a de bien meilleures chances d'être applicable que  $B2$ . Cette indication peut être utilisée par un système de sélection de l'action à un niveau plus élevé, celui-ci pouvant la combiner avec d'autres sources d'informations (tâche en cours, priorités, etc ...). Ainsi l'architecture de sélection de l'action EMF [Drogoul, 1993], portée sur ces mêmes robots, utilise des stimuli pour déclencher les comportements. La valeur de  $U_B(X)$  pour un comportement peut, dans ce cadre, être assimilée à un tel stimuli.

## 12.5 Conclusion

Un comportement, une fois appris, doit pouvoir être utilisé dans une application plus large. Le modèle **MPL** fournit au travers de la mesure  $U_B(X)$  un moyen simple pour contrôler ce comportement et permettre son *intégration* dans un système complet. Le mécanisme de perception active, qui repose en grande partie sur cette mesure d'utilité, est utilisé dans les expérimentations d'acquisition de comportements décrites dans le chapitre 15.

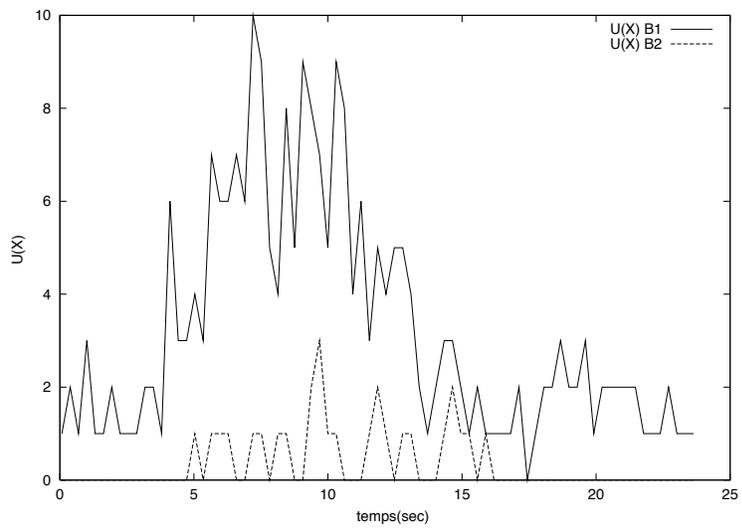


FIG. 12.7: *Mesure d'utilité pour deux comportements B1 et B2 appris dans des contextes différents. La mesure effectuée dans l'environnement de B1 est nettement en défaveur de B2.*

## Chapitre 13

# Structuration du champ perceptuel

### 13.1 Introduction

Comme il a été dit dans le paragraphe 9.2.1, le champ perceptuel est l'intermédiaire par lequel le robot perçoit son environnement. Il est formé de plusieurs plans, sur chacun desquels est projetée une propriété perceptuelle déduite des données sensorielles. La structuration du champ perceptuel consiste à définir les propriétés perceptuelles utilisées en fonction de l'environnement dans lequel le robot doit évoluer (on ne prendra probablement pas les mêmes propriétés pour un robot devant évoluer dans des bureaux et un autre situé dans une exploitation agricole), elle doit permettre d'apprendre des comportements différents pour un même environnement. C'est à l'occasion de cette phase de structuration que le concepteur peut introduire dans le système un *biais perceptuel* donnant au robot une chance d'utiliser son environnement efficacement, le mécanisme d'apprentissage, quel qu'il soit, restant tributaire de ces biais pour avoir une chance de généraliser à partir des données des démonstrations. Heureusement, le concepteur n'est pas condamné à définir à chaque fois des propriétés perceptuelles ad-hoc car certaines d'entre elles se montrent plus efficaces que d'autres pour une classe d'environnement (ie : les locaux de bureau). Déterminer ces propriétés et définir leur condition d'application constitue en quelque sorte *une pragmatique de la perception en vue de l'action* qui est à mon sens indispensable en robotique.

Il est envisageable d'apprendre ces propriétés, comme le suggère *l'apprentissage perceptuel* en psychologie [Goldstone, 1998] et comme c'est le cas dans plusieurs travaux portant sur l'apprentissage en robotique [Bredeche *et al.*, 2002] [Klingspor *et al.*, 1996]. Cependant, ici

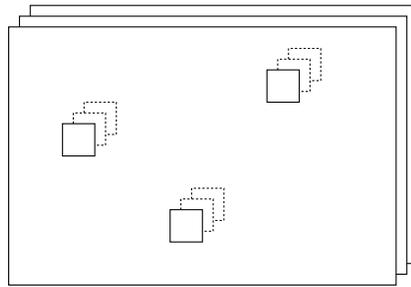


FIG. 13.1: Champ perceptuel décomposé en plan perceptuels

il y a peu de critères pour déterminer *la valeur* d'une propriété par rapport à une autre et l'espace des propriétés possibles est infini (espace des fonctions sur les données sensorielles). Pour cette raison j'ai choisi de n'utiliser que des propriétés perceptuelles locales dont la portée est limitée à de petites zones du champ perceptuel (hypothèse de tessellation cf. section 7.3.2 et ci-dessous). Ces propriétés ont été définies expérimentalement en comparant l'impact de chacune sur la qualité de l'acquisition des comportements.

Les propriétés perceptuelles envisageables sont les propriétés locales qui permettent de distinguer un point du plan perceptuel de ses voisins; elles peuvent correspondre à une densité locale, une intensité ou encore une variation (fig. 13.1). On peut retrouver une certaine correspondance entre cette structure en plans et la projection des terminaisons sensorielles de l'épiderme et des champs récepteurs de la rétine sur des aires sensorielles à deux dimensions telle qu'elle est décrite par les neuro-biologistes ( par exemple les cartes rééchantillonnées de Gerald Edelman [Edelman, 1992]). Dans le modèle proposé cette structure est évidemment idéalisée et très simplifiée. J'ai cependant voulu insister sur trois aspects transposés des systèmes biologiques. La **tessellation** suppose que la perception est limitée à des micro-percepts ne portant chacun que sur une toute petite partie du champ perceptuel (champs récepteurs) et décrivant celle-ci avec *un vocabulaire réduit*. Il s'agit d'un processus *d'identification* d'une zone locale (concluant par exemple que cette zone 'est un point rouge') contrairement au processus d'un réseau de neurones qui appliquerait une fonction à cette même zone. **L'homologie de forme** entre le stimulus externe et la représentation interne permet d'éviter d'incorporer dans le robot un modèle géométrique du monde. Les relations topologiques telles que 'à droite de', 'à côté', 'au dessus de' sont conservées de manière implicites. **La séparation en plusieurs voies perceptuelles** (couleur, proto-forme etc.) permet d'associer plusieurs visions simultanées et

complémentaires de l'environnement.

Pour le cas du robot élémentaire, le champ perceptuel est limité au champ du senseur visuel et la population de cellules conserve une organisation rétinotopique organisée en fonction du référentiel rétinien de l'image vidéo. Par extrapolation, on peut imaginer une version totale du champ perceptuel pour un robot beaucoup plus élaboré. Un tel champ perceptuel ferait coïncider des plans perceptuel correspondant à des entrées sensorielles aussi diverses qu'un épiderme ou un capteur visuel étendu (stéréoscopique, omnidirectionnel, avec rémanence, etc.). Ces plans ne serait pas limité à un champ visuel mais s'étendrait à tous le 'corps' du robot par intégration temporelle des données sensoriel.<sup>26</sup>

## 13.2 Quelles propriétés perceptuelles ?

Dans la suite du chapitre on ne considère que les données sensorielles visuelles du robot élémentaire (RE). Celles-ci se résument à une matrice donnant pour chaque point l'intensité des trois composantes rouge, verte et bleue (eq .13.1). Le problème concret traité ici est donc le passage de cette matrice  $X$  issue de la caméra aux plans perceptuels.

$$X = \begin{bmatrix} [rgb] & [rgb] & \cdots & [rgb] \\ [rgb] & [rgb] & \cdots & [rgb] \\ \cdots & \cdots & \cdots & \cdots \\ [rgb] & [rgb] & \cdots & [rgb] \end{bmatrix} \quad (13.1)$$

Les techniques utilisées en vision sont extrêmement nombreuses et on trouvera une description détaillée des plus fréquemment utilisée dans [Watt and Policarpo, 1998] et [Sonka *et al.*, 1993]. Parmi les plus communes :

- Les techniques appliquées pour la *préparation* des images : égalisation, normalisation, filtrage, convolution.
- celles venant ensuite pour leur 'interprétation' : en particulier la détection de contour

---

<sup>26</sup>Des faits neurobiologiques plaident en faveur de cette possibilité , ainsi Alain Berthoz décrit des neurones bimodaux découverts chez le singe : "*Ces neurones bimodaux ont des propriétés fascinantes : par exemple, si un neurone décharge lorsqu'on touche la joue du singe, le même neurone déchargera aussi si l'on approche la main de la joue sans le toucher. Le "champ récepteur " spatial visuel correspond au "champ récepteur tactile". On peut ainsi déclencher l'activité du neurone en plaçant la main ou un objet dans un volume d'environ dix centimètres autour de la joue.*"

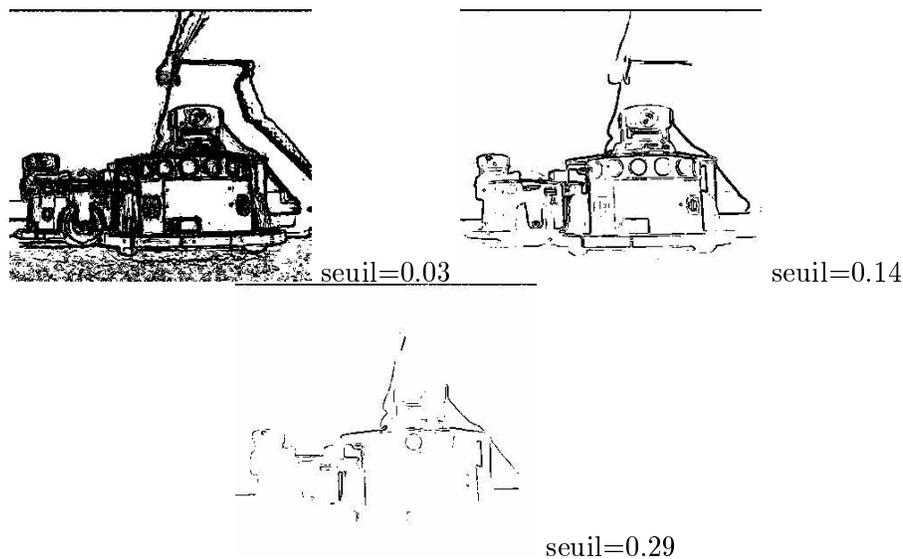


FIG. 13.2: Les transformations d'images sont très sensibles au paramétrage. Cas d'une détection de contour avec un opérateur de Sobel puis binarisation avec des seuils différents.

et la segmentation de l'image en zones significatives.

J'ai choisi, dans ce travail, de ne pas utiliser de techniques classiques de détection de contour ni de segmentation. D'une part, ces méthodes s'inscrivent avant tout dans des schémas, éventuellement coûteux, de reconnaissance d'objets. D'autre part, elles impliquent d'effectuer une recherche dans le champ visuel car elles concentrent l'information sur peu de points : la figure 13.2 montre le résultat de l'application d'un opérateur de détection de contour sur une image, où seuls quelques points sont porteurs d'information. Enfin, elles sont particulièrement sensibles au paramétrage et doivent à chaque fois être choisies minutieusement en fonction du type d'images à traiter [Pal and Pal, 1993].

La figure 13.2 montre le cas classique d'une détection de contour par un opérateur de Sobel et par binarisation des pixels en deux classes (noir ou blanc) selon un seuil. Selon la valeur de seuil choisie, l'image résultante change complètement. Bien que la détection de contour ait été invoquée comme étant un mécanisme essentiel de la vision [Marr, 1982], je ne suis pas convaincu qu'elle doive être utilisée systématiquement et explicitement dans un modèle de vision robotique. D'ailleurs, les micro-percepts reposant sur deux pixels dans la variante **MPL-s** sont une sorte de proto-détecteur de contour. La segmentation est encore plus sujette à caution dans un environnement dynamique où le robot doit réagir rapidement. La figure 13.2



FIG. 13.3: A gauche, une configuration de carrés colorés dans cinq conditions d'illumination. Chaque image comporte un même carré vert marqué par un rectangle. A droite, détection du seul carré vert avec un algorithme de segmentation robuste.

(a) montre une configuration de carrés colorés dans cinq conditions d'illumination différentes. La figure (b) montre l'image (a) après application d'un algorithme de segmentation identifiant des zones de couleur par classification des pixels. Pour ce test, j'ai utilisé un algorithme robuste proposé par J. Bruce [Bruce *et al.*, 2000] et conçu spécifiquement pour les robots de la RoboCup. On fournit à cet algorithme des plages de couleurs (définies dans l'espace teinte saturation valeur) et les zones correspondant à ces classes de couleurs sont ensuite identifiées dans l'image. Comme on peut le voir sur la figure 13.2 (b) un même carré vert n'est pas toujours détecté.

Comme j'en ai déjà fait mention dans la section 7.2.3 la neurobiologie distingue fréquemment trois canaux principaux dans le système visuel des mammifères : la couleur, la forme élémentaire et la direction du mouvement. Plutôt que d'utiliser des mécanismes de 'haut niveau' comme la segmentation, j'ai choisi de reprendre deux de ces trois canaux élémentaires : la couleur et la forme élémentaire (ou proto-forme). Ce choix n'est pas uniquement motivé par la correspondance avec les voies identifiées par la biologie, mais aussi par le fait que les environnements humains (bureaux, usines, etc.) comportent beaucoup d'informations codées sous forme de couleur (signalétique, ameublement, etc.) ou de formes élémentaires (coins, bords, transitions) et que l'extraction de ces informations est peu coûteuse et compatible avec des moyens de calculs embarqués. Enfin, ces informations restent locales par rapport à l'image et sont facilement utilisables dans un modèle distribué tel que **MPL**. L'information

de mouvement qui sera certainement intéressante à intégrer à terme, en particulier pour des comportements nécessitant d'interagir avec d'autres robots, n'est pas prise en compte pour le moment.

### 13.3 Constitution des plans perceptuels

Les informations extraites du flot visuel doivent être suffisamment robustes aux diverses conditions environnementales. Pour cette raison, le passage des données sensorielles aux plans perceptuels est réalisé par une étape de *filtrage* suivie par une étape de *classification* selon le schéma de la figure 13.4.

- L'étape de filtrage a pour rôle principal de diminuer l'impact des changements de luminosité et d'homogénéiser les données visuelles.
- L'étape de classification est chargée d'extraire des données filtrées, et de façon robuste au bruit, les valeurs qui seront mémorisées dans un plan perceptuel.

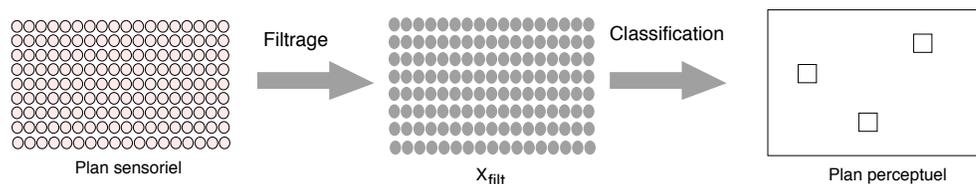


FIG. 13.4: Des données sensorielles au plan perceptuel

#### 13.3.1 Filtres

Les filtres sont appliqués successivement à l'image de départ  $X$  pour obtenir l'image filtrée  $X_{filt}$ .

$$X_{filt} = Filtre_n(\dots Filtre_2(Filtre_1(X)) \dots) \quad (13.2)$$

Les principaux filtres utilisés représentent des prétraitements classiques employés pour diminuer l'impact des variations de l'intensité lumineuse et lisser les images. Ils sont récapitulés ci-dessous.

- Filtre norm1 : normalisation des valeurs  $r$ ,  $g$ ,  $b$  par rapport à la valeur  $max(r, g, b)$  relevée dans l'image.

- Filtre norm2 : normalisation des valeurs r, g, b, séparément chacune par rapport à  $max(r)$ ,  $max(g)$ ,  $max(b)$  relevées dans l'image.
- Filtre norm3 : normalisation des valeur r, g, b par rapport aux trois valeurs moyennes  $\bar{r}$ ,  $\bar{g}$ ,  $\bar{b}$ . (normalisation de type 'Grey World').
- Filtre norm4 : normalisation de valeurs r,g,b par rapport à la somme  $r + g + b$ .
- Filtre smooth1 : convolution de l'image avec un noyau gaussien de 5 par 5 pixel (gaussian blur).
- Filtre smooth2 : moyenne; chaque pixel de l'image devient la moyenne de ses 9 voisins.
- Filtre smooth3 : pixelisation; les pixels sont regroupés par pavés de quatre ou huit pixels et prennent pour valeur r,g,b la moyenne du groupe.

D'autres filtres ont été essayés (cf. 14.4.3.0) mais n'ont pas été conservé compte tenu des artefacts qu'ils généraient dans l'image, particulièrement à basse résolution.

### 13.3.2 Classeurs

Les valeurs mémorisées dans un plan perceptuel (le plan Forme par exemple) sont obtenues par classification des zones de l'image  $X_{filt}$ .

$$Plan_{Forme} = \begin{bmatrix} classe & classe & \cdots & classe \\ classe & classe & \cdots & classe \\ \cdots & \cdots & \cdots & \cdots \\ classe & classe & \cdots & classe \end{bmatrix} \quad (13.3)$$

A chaque plan correspond un classeur particulier. Un classeur est local, il fait correspondre une classe discrète à un voisinage (ou champ récepteur) centrée sur une position  $(xpos, ypos)$  de l'image (figure 13.5 et équation 13.4).

$$Forme(xpos, ypos) = classeur_{Forme}(voisinage_{X_{filt}}(xpos, ypos)) \quad (13.4)$$

Trois types de classeurs ont été envisagés :

- Classeur de type forme : shape1.
- Classeur de type couleur : hsv431, hsv332, hsv333, colorclass, lumin32, lumin128.
- Classeur de type contour : edge1.

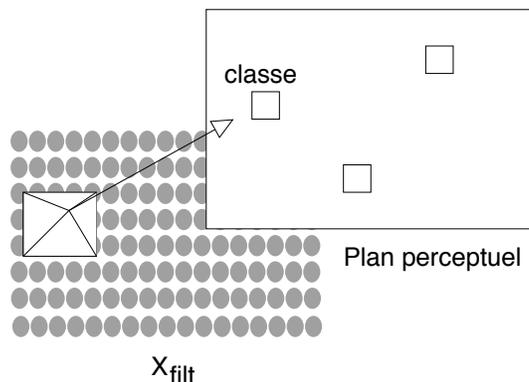


FIG. 13.5: Classeur local

### Classeur Forme

Le classeur local pour la forme **shape1** est conçu pour discriminer des formes très simples ou proto-formes. Ce classeur a été obtenu par apprentissage comme suit : une série d'images a été récoltée dans le laboratoire par le robot. Celles-ci ont été converties en niveau de gris et découpées en zones de  $5 \times 5$  pixels. Cet ensemble de vignettes a été ensuite utilisé pour entraîner un classeur auto-organisé (SOMF) de Kohonen [Kohonen, 1997] comportant  $6 \times 5 = 30$  classes. On rappelle qu'un classeur de type SOMF permet de projeter un ensemble de vecteurs de grande dimension (ici  $5 \times 5 \times 3$ ) dans un espace à deux dimensions (ici  $6 \times 5$ ) en conservant les relations topologiques entre les vecteurs (proximité d'un vecteur à un autre). Par la suite, classer un nouveau vecteur revient à trouver le point de l'espace à deux dimensions dont il est le plus proche. Les vecteurs obtenus après l'apprentissage SOMF et correspondant à chacune des trente classes sont représentés dans la figure 13.6, un vecteur étant visualisé par une vignette de  $5 \times 5$  pixels en niveau de gris.

Ce classeur est ensuite utilisé pour classer un point  $p$  de l'image en comparant son voisinage de  $5 \times 5$  pixels avec chacun des 30 vecteurs de classe. Le vecteur de classe le plus proche (au sens de la distance euclidienne) donne sa classe au point  $p$  (eq. 13.5). L'apprentissage du classeur Forme est effectuée une fois pour toutes et le même classeur est réutilisé pour tous les comportements.

$$Forme(xpos, ypos) = \underset{c \in [1..30]}{\operatorname{argmin}} \|voisinage(xpos, ypos) - VecteurClasse_c\| \quad (13.5)$$

L'utilisation d'un classeur de type SOMF est intéressante ici pour deux raisons :

- Il permet de construire facilement un lexique de 30 proto-formes (fig. 13.6) à partir de données issues de l'environnement du robot. Le lexique couvre toute les situations rencontrées (dans l'environnement de bureau que constitue le laboratoire).
- La classification est peu sensible au bruit, des faibles variations sur une forme ne changent pas sa classe.

### Classeur Couleur

Les différents classeurs utilisés pour constituer un plan Couleur : colorclass - hsv432, hsv332, hsv333 - lumin32, lumin128.

**Le classeur colorclass** reprend le principe de la classification de Kohonen utilisée pour shape1. L'ensemble d'apprentissage est dans ce cas constitué des triplets r, g, b correspondant à des pixels provenant d'images enregistrées aléatoirement dans l'environnement. Le classeur permet d'associer une classe parmi  $15 = 5 \times 3$  à chaque pixel de l'image  $X_{filt}$  en cherchant le vecteur le plus proche.

**Les classeur hsv431, hsv332, et hsv333** associent une classe au triplet (r,g,b) décrivant un pixel. La couleur du pixel est convertie dans l'espace HSV.<sup>27</sup> Le codage HSV est classique en vision et permet d'agir séparément sur la luminosité ou la saturation d'une couleur donnée; la conversion (r,g,b)-> (h,s,v) utilisée est reprise de [Foley *et al.*, 1995]. La discrétisation des trois valeurs h,s,v est effectuée en les représentant respectivement avec  $h_q$  bits,  $s_q$  bits et  $v_q$  bits. Il est ainsi possible d'accentuer l'importance de la teinte par rapport à la valeur et donc de moins prendre en compte les variations de luminosité. Les différentes valeurs utilisés pour la quantification sont décrites dans la table 13.1 et le nombre de classes reconnues par un classeur est égal à  $2^{h_q+s_q+v_q}$ . Le classeur hsv431 donne par exemple beaucoup d'importance à la teinte et à la saturation mais très peu à la luminosité : les couleurs correspondant à ses 256 classes sont visibles dans la figure 13.7.

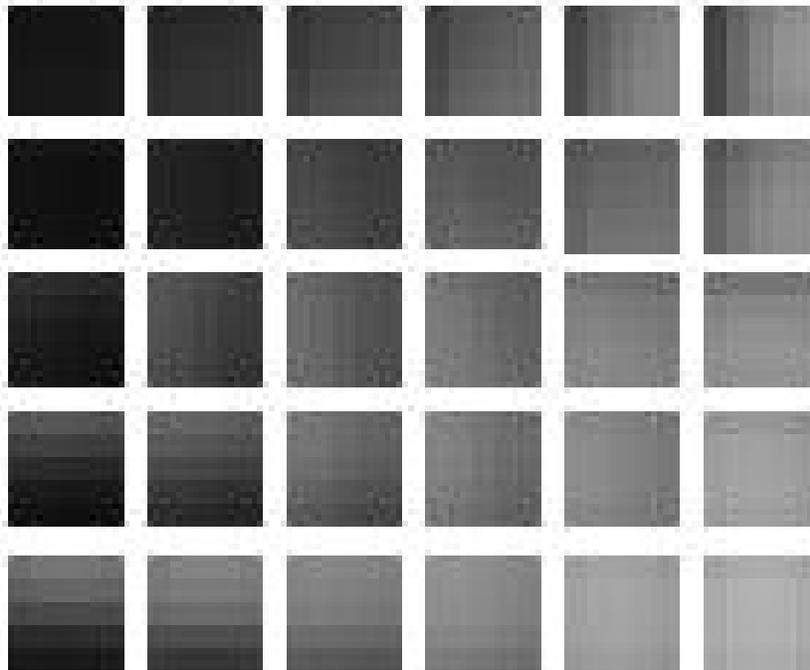
---

<sup>27</sup>Dans l'espace HSV une couleur est représentée par trois valeurs, sa teinte (Hue), la saturation plus ou moins forte de cette teinte (Saturation) et la luminosité de la couleur (Value)

	$h_q$	$s_q$	$q_q$	nclasses
hsv431	4	3	1	256
hsv332	3	3	2	256
hsv333	3	3	3	512

TAB. 13.1: Trois types de discrétisation pour classer les couleurs.

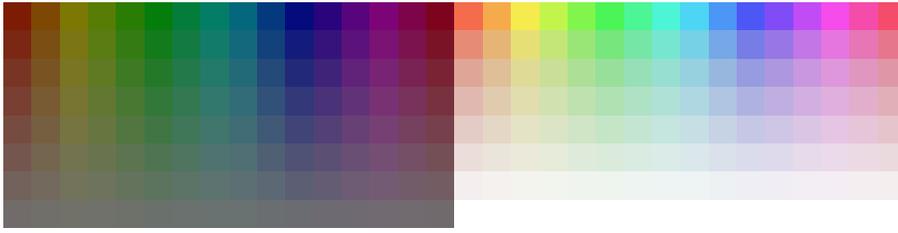
Les classeurs **lumin32** et **lumin128** sont les plus simples et correspondent à la valeur discrétisée du pixel en niveaux de gris.<sup>28</sup> Lumin32 dicrétise le niveau de gris sur 32 valeurs et lumin128 sur 128 valeurs.

FIG. 13.6: Les  $6 \times 5 = 30$  classes distinguées par le classeur *shape1*. Une vignette de  $5 \times 5$  pixels issue de l'image  $X$  sera comparée à ces 30 vignettes, et prend la classe de la plus proche.

### Classeur Contour

Le classeur **edge1** associe une classe à tout voisinage  $3 \times 3$  de l'image  $X_{filt}$ . Il utilise deux masques (cf. [Sonka *et al.*, 1993]) Hor et Vert pour déterminer un gradient dans l'image et un masque Moy pour effectuer une moyenne sur le voisinage. Une image  $I$  est d'abord calculée conformément à l'équation 13.6 (où  $\otimes$  est l'opérateur de convolution de l'image avec

<sup>28</sup>La conversion en niveau de gris utilise la formule  $gray(r, g, b) = 0.29r + 0.58g + 0.11b$ .

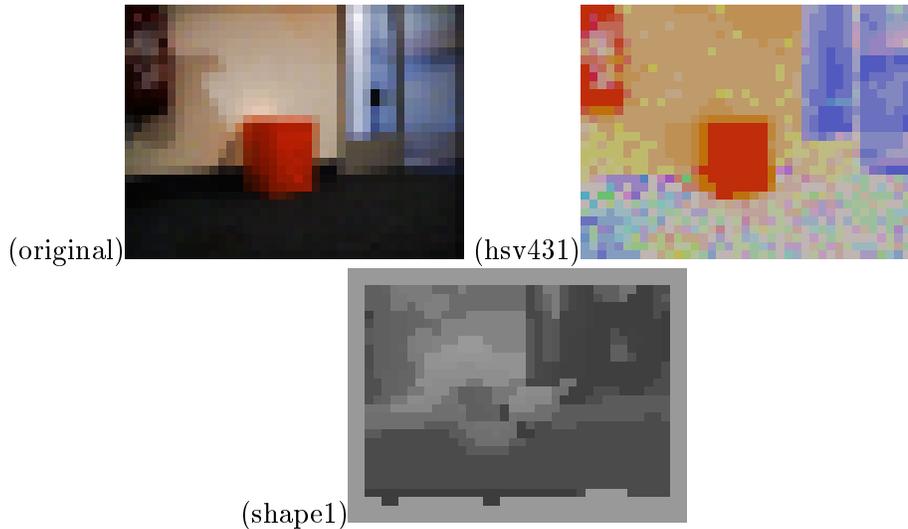
FIG. 13.7: 256 classes de couleurs distinguées par le classeur *hsv431*.

un masque  $3 \times 3$ ). Puis la valeur  $I(xpos,ypos)$  est discrétisée sur 4 niveaux pour obtenir la classe.

$$I = Moy \otimes \max(|Hor \otimes X_{filt}|, |Vert \otimes X_{filt}|) \quad (13.6)$$

$$Hor = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad Vert = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad Moy = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad (13.7)$$

### 13.3.3 Conclusion

FIG. 13.8: Visualisation en fausses couleurs de la classification (*hsv431*) et (*shape1*) appliquée à l'image (originale)

Trois étapes ont été nécessaires pour déterminer expérimentalement la structure du champ perceptuel. Une première étape a consisté à essayer différentes combinaisons de filtres et de classeurs sur des fichiers démonstrations grâce à la plate-forme d'apprentissage (section 14.4.3.0) et d'étudier les effets des différents codages (suppriment-ils de l'information ? homogénéisent-ils des images similaires ?) ; cette étape a eu en particulier pour effet de supprimer les filtres et

classieurs créant trop d'artefacts dans l'image. Une deuxième étape a consisté à tester l'impact de différentes combinaisons de filtres et de classeurs sur l'apprentissage en utilisant une mesure objective (MSE), une partie de ces expérimentations étant présentée dans la section 15.2. Enfin la troisième étape a consisté à conserver les configurations permettant effectivement d'apprendre et de restituer des comportements.

Parmi les configurations passant les trois étapes la configuration suivante à été retenue :

- Filtre norm2
- Plan Couleur obtenu grâce au classeur hsv431.
- Plan Forme obtenu grâce au classeur shape1.

La détermination expérimentale de cette configuration comporte une part importante d'arbitraire, mais il ne peut en être autrement compte tenu du nombre de filtres et de classeurs envisageables. Pour palier ce caractère arbitraire, il sera intéressant à terme de discerner les couple (configuration, environnement) permettant d'apprendre des comportement et de constituer peu à peu ce que j'appelais dans le début du chapitre une pragmatique de la perception en vue de l'action.

Troisième partie

**Expérimentations**



# Chapitre 14

## Plate-forme expérimentale

### 14.1 Introduction

Les précédents chapitres sont restés très indépendants de toute plate-forme matérielle. Comme il importe dans une phase d'expérimentation de préciser les moyens utilisés, particulièrement lorsqu'il s'agit d'expérimentations en robotique, ce chapitre présente donc brièvement le dispositif ayant servi de support aux expériences d'acquisition de comportements. Il décrit d'abord la plate-forme robotique, puis l'implantation qui y est faite du robot élémentaire, et enfin la plate-forme d'apprentissage utilisée par le tuteur. La mise en oeuvre logicielle est succinctement décrite à la fin du chapitre.

### 14.2 Plate-forme robotique



FIG. 14.1: Robot Pioneer 2DX utilisé dans la phase expérimentale.

Le robot utilisé dans la phase expérimentale est un robot de type Pioneer 2DX fabriqué par la société américaine ActivMedia (figure 14.1). Ce modèle est, avec les robots Koala et Nomad, parmi les plus utilisés dans les laboratoires de robotiques.

### 14.2.1 Senseurs

Le robot est équipé de différents senseurs : une *caméra*, des *sonars*, un *odomètre* et des *bumpers* décrits ci-dessous.

**Caméra :** la caméra est une caméra couleur monoscopique CCD de type Sony EVI D31 (fig. 14.2). Elle fournit des images d'une résolution maximale de  $640 \times 480$  pixels. Les images sont obtenues grâce à une carte d'acquisition embarquée sur le robot. La caméra est orientable verticalement (tilt) et horizontalement (pan) et dispose d'un zoom.

**Sonars :** 16 sonars à ultrason de type polaroïd équipent le robot (fig. 14.3). Ils sont organisés autour du robot tous les 15 degrés. Ils permettent d'obtenir la distance à l'obstacle le plus proche dans la seule direction pointée par le sonar et ceci avec une précision de  $\pm 20cm$ . Au-delà de 5 mètres la mesure n'est plus significative. La mesure est mise à jour 3 fois par seconde.

**Odomètre :** le robot dispose d'un mécanisme odométrique qui lui permet de déduire approximativement sa position relative du robot par rapport à un point de départ. Ce déplacement relatif est obtenu grâce à des encodeurs situés sur les moteurs. Comme tous les mécanismes odométriques, il est très imprécis et accumule les erreurs.

**Bumpers :** huit bumpers sont installés autour du robot. Chacun correspond à un indicateur de contact de type tout ou rien signalant lorsque le robot heurte un objet.

### 14.2.2 Effecteurs

Les seuls effecteurs de ce robot sont ses deux roues motrices et la tête mobile de la caméra décrite ci-dessus. Les roues sont mues par deux moteurs indépendants permettant un contrôle des mouvements de type différentiel. Les deux roues sont contrôlables dans les limites suivantes : vitesse maximale de translation  $600mm/sec$ , vitesse maximale de rotation  $150deg/sec$ .



FIG. 14.2: Camera couleur monoscopique



FIG. 14.3: Ceinture de sonars

### 14.2.3 Moyens informatiques embarqués

Les moyens informatiques embarqués sur le robot sont *un ordinateur de type PC, un microcontrôleur et une liaison radio.*

**PC embarqué** : l'ordinateur embarqué est de type PC/linux et dispose d'un processeur Intel MMX cadencé à 200Mhz ainsi que de 32 Mo de mémoire.

**Liaison Radio Ethernet** : le PC est relié au réseau local du laboratoire par une liaison ethernet-radio de type IEEE 803.11b autorisant une bande passante d'environ 10mbits. Toutes les transactions s'effectuent en utilisant les protocole TCP/IP et UDP.

**Carte Microcontrôleur** : Le robot dispose aussi d'une carte Microcontrôleur doté d'un processeur MC68HC11 chargé des échanges de bas niveau entre les PC et les senseurs/effecteurs.

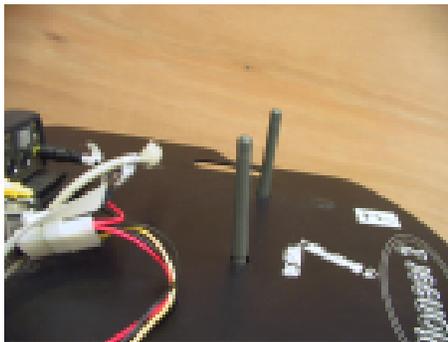


FIG. 14.4: Liaison radio ethernet

### 14.3 Implémentation du robot élémentaire (RE)

La mise en oeuvre du robot élémentaire décrit dans la section 8.2 sur le robot réel n'a pas posé de problèmes particuliers. Rappelons que le robot élémentaire suppose de définir :

- une matrice réelle  $X$  donnant la couleur de chaque point du champ visuel.
- un vecteur réel  $Y = [Y_{left}, Y_{right}]^T$  pour le contrôle des mouvements.

Le vecteur effecteur  $Y$  est constitué de deux valeurs  $Y_{left}$  et  $Y_{right}$  correspondant à la vitesse de chacune des deux roues. Chaque vitesse exprimée en millimètres par secondes est une valeur réelle dans l'intervalle  $[-600mm/s, 600mm/s]$ . En pratique la vitesse de  $250mm/s$  ( $\approx 1$  km/heure) n'est jamais dépassée.

La matrice sensorielle  $X$  est obtenue à partir de l'image issue de la caméra. L'image initiale est au format  $320 \times 240$  pixels, elle est réduite à une résolution de  $40 \times 30$ . Chaque pixel est décrit par ses 3 composantes rouge, vert, bleu à valeurs entières dans l'intervalle  $[0, 255]$ . La réduction est réalisée par pixelisation c'est à dire en faisant la moyenne de 16 pixels voisins de l'image originale. Finalement la matrice  $X$  est constituée de  $40 \times 30$  vecteurs  $r, g, b$  comme indiqué dans l'équation 14.1.

$$X = \begin{bmatrix} [rgb]_{0,0} & [rgb]_{1,0} & \cdots & [rgb]_{39,0} \\ [rgb]_{0,1} & [rgb]_{1,1} & \cdots & [rgb]_{39,1} \\ \cdots & \cdots & \cdots & \cdots \\ [rgb]_{0,29} & [rgb]_{1,29} & \cdots & [rgb]_{39,29} \end{bmatrix} \quad (14.1)$$

Les informations de  $Y$  et  $X$  sont mis à jour toutes les 330 millisecondes soit approximativement à une fréquence de 3 hz. Les démonstrations enregistrées contiennent de ce fait 3 frames par

seconde. Le choix de cette fréquence, ainsi que de la résolution, est une conséquence, bien évidemment, de la faible puissance des moyens informatiques embarqués.

## 14.4 Plate-forme d'apprentissage

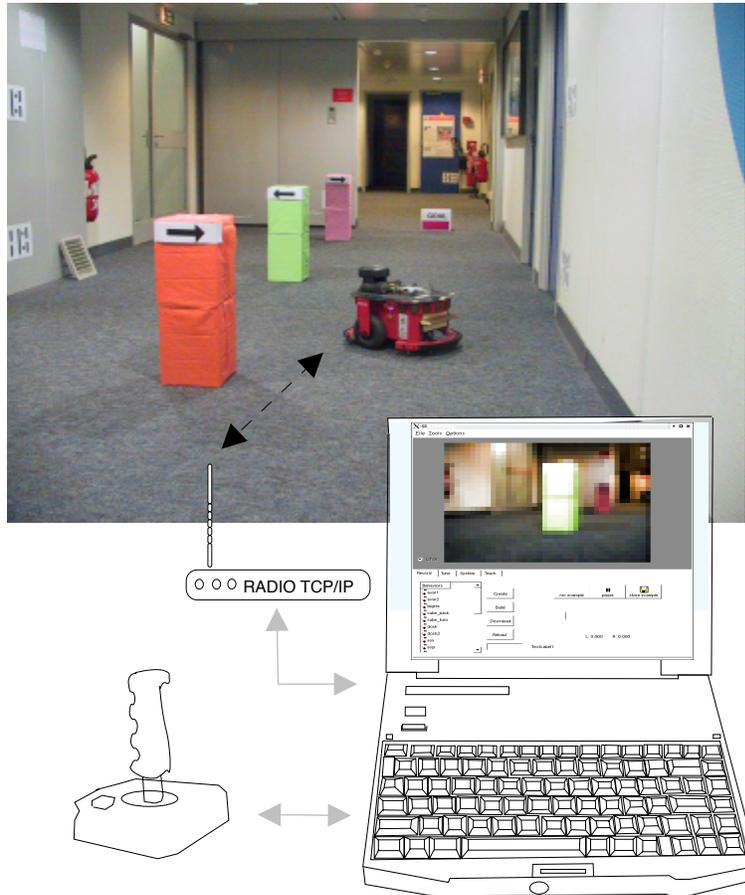


FIG. 14.5: Plate-forme d'apprentissage utilisée par le tuteur.

La figure 14.5 représente la plateforme d'apprentissage utilisée dans toutes les expérimentations. Celle-ci se compose d'un ordinateur portable connecté au réseau sans-fil. Un joystick analogique précis à deux axes permet de contrôler les mouvements du robot pendant les démonstrations. L'image vidéo provenant du robot est retransmise sur l'écran du tuteur en temps réel, permettant ainsi à celui-ci de voir ce que perçoit le robot.

### 14.4.1 Fichiers de démonstration

A partir de cette plate-forme le tuteur enregistre les fichiers de démonstration. A titre indicatif, la taille d'un fichier de démonstration, qui dépend de la durée, est d'approximativement 750 kilo-octets pour chaque minute. Ils sont conservés dans une base de données localisée sur l'ordinateur portable.

### 14.4.2 Pilotage à distance

Lors des démonstrations, la loi de contrôle des deux roues  $Y_{left}$ ,  $Y_{right}$  du robot est obtenue par rotation et changement d'échelle à partir des valeurs  $Axe_{vert}$ ,  $Axe_{hor}$  provenant du joystick (fig. 14.6). Les commandes de mouvements ne sont ni filtrées ni discrétisées. De ce fait, le joystick permet un contrôle très précis des mouvements du robot en utilisant toute la gamme de valeurs disponible.

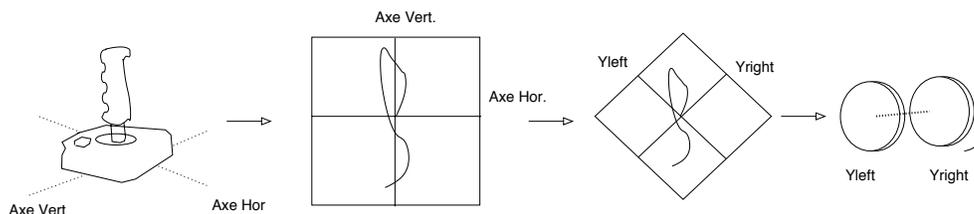


FIG. 14.6: Loi de contrôle du robot à partir du joystick.

### 14.4.3 Interface utilisateur

L'interface utilisateur permet au tuteur d'enregistrer des démonstrations, de déclencher la synthèse d'un comportement, d'activer sa réalisation autonome par le robot et de monitorer ce dernier. Elle se présente sous la forme d'une fenêtre principale (fig. 14.7) et de plusieurs boîtes de dialogue. La fenêtre principale montre en particulier le flot vidéo capturé par la caméra du robot et propose une interface simple de type magnéscope pour gérer les démonstrations.

### Analyse des démonstrations

L'interface fournit par ailleurs un outil de visualisation et d'analyse des films de démonstrations enregistrés (fig. 14.8). L'outil permet de dérouler les films pas à pas ou à vitesse normale et d'appliquer à ceux-ci différents filtres et pré-traitements. Il intègre une version de l'algorithme *Realize* identique à celle implantée sur le robot et permet de comparer après

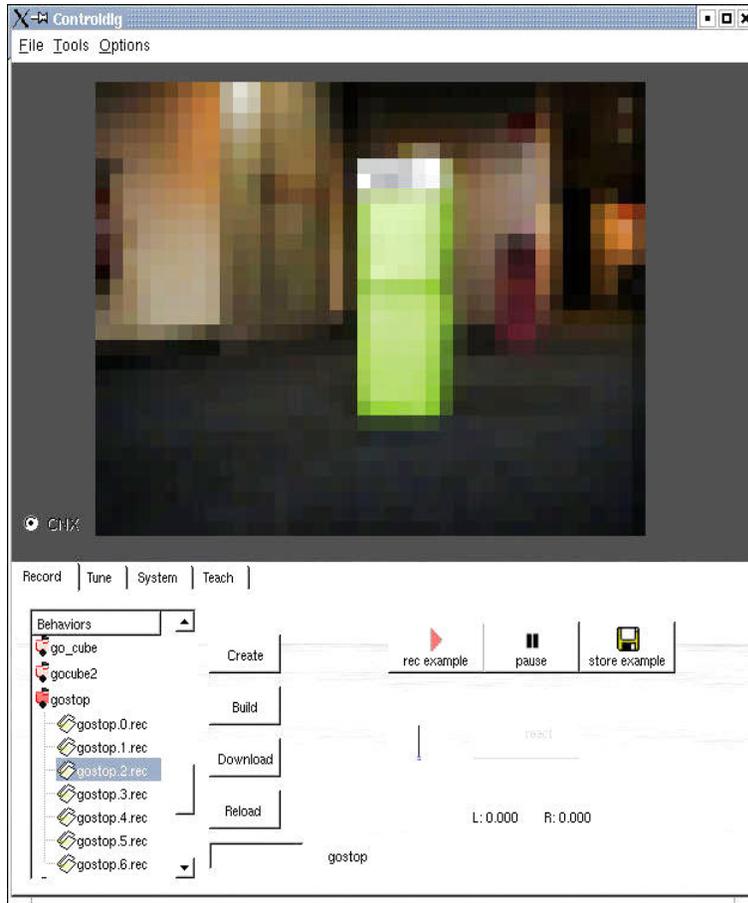


FIG. 14.7: Interface utilisateur pour l'enregistrement de démonstrations.



FIG. 14.8: Interface utilisateur - Outil pour la visualisation et l'analyse des films de démonstrations. Ici, un filtre de type sobel 3 est appliqué sur le film (à droite). L'image originale est visible à gauche.

Filtre blobs	Filtre normalize r/r+g+b
Filtre normalize grey world	Filtre normalize r/max(r,g,b)
Filtre normalize (sep) r/max(r)	Filtre hue -> gray
Filtre sat -> gray	Filtre val -> gray
Filtre pixelize 2x2	Filtre smooth
Filtre gaussian Blur	Filtre quantize rgb
Filtre filterHsv	Filtre quantizeHsv(1)
Filtre quantizeHsv(2)	Filtre local density
Filtre local density ->gray	Filtre median filter (indexed)
Filtre gray	Filtre luminance 16 levels
Filtre luminance 32 levels	Filtre luminance 64 levels
Filtre luminance 256 levels	Filtre stripAverage
Filtre stripMaxLuminance	Filtre stripMinLuminance
Filtre laplace	Filtre sobel1
Filtre sobel2	Filtre sobel3
Filtre edge detect(xv)	Filtre temporal difference
Filtre temporal mean	Filtre quantizehsv X edge detect(laplace)
Filtre quantizeHsv (pixcells)	Filtre quantizeHsv (pixcells) fake colors
Filtre Cells Saliency	Filtre Cells Credit
Filtre Cells Gain	Filtre Shape PerceptClassifier
Filtre Color PerceptClassifier	Filtre Online Reactions

TAB. 14.1: Filtres combinables dans l'outils d'analyse ShowView

apprentissage, et hors-ligne, la réaction du robot à telle ou telle situation. Cet outil s'est avéré extrêmement utile dans la phase d'étude, en particulier pour expérimenter et déterminer les biais perceptuels les plus intéressants. Les différents filtres (applicables à l'image  $X$ ) implantés sont donnés table 14.1.

## 14.5 Mise en oeuvre logicielle

Pour pouvoir expérimenter le modèle **MPL** sur la plateforme décrite dans les sections précédentes, il m'a été nécessaire de réaliser deux applications *MPL-Learn* et *MPL-Teach* localisées respectivement sur le robot et sur le portable du tuteur. On retiendra qu'une approche

---

temps réel (identification des activités critiques, division en threads distincts et synchronisation) est ici indispensable pour une mise en oeuvre effective du modèle. Cette approche est en particulier nécessaire pour permettre une boucle perception-action suffisamment rapide s'exécutant de manière identique lors de l'enregistrement de démonstrations et lors de la réalisation autonome, ainsi que pour une prise en compte rapide des commandes du tuteur et de leur feedback.

### 14.5.1 Environnement logiciel

L'environnement logiciel des deux applications est schématisé dans la figure 14.9. La plateforme robotique est livrée par la société ActivMedia avec deux composants logiciels : SAPHIRA et P2OS. P2OS constitue une abstraction de premier niveau des senseurs et effecteurs du robot (hormis la caméra). SAPHIRA est supposé fournir un environnement de programmation ouvert adapté aux applications robotiques. Malheureusement, et comme c'est souvent le cas avec les packages accompagnant les robots, celui-ci est très sous-dimensionné : pas de prise en compte de la vision, contrôle temps réel mal adapté et trop restreint, limitations nombreuses pour l'architecture de l'application. De ce fait, ce composant n'est utilisé que pour transmettre quelques commandes de bas niveau (en particulier la commande de vitesse correspondant à chaque roue) à la partie micro-contrôleur gérée par P2OS. L'application MPL-Learn prend en charge quant à elle toutes les autres fonctionnalités, en particulier, la vision, la boucle perception-action, la communication, le monitoring et l'ordonnancement des activités.

L'application MPL-Teach tourne sur l'ordinateur distant et est principalement dédiée à l'interface utilisateur de la plateforme d'apprentissage. Pour des raisons de performances, elle repose sur la bibliothèque graphique QT 2.0. J'ai réalisé les deux applications MPL-Learn et MPL-Teach en C++ sous Linux, leur poids cumulé étant approximativement de 35.000 lignes.

### 14.5.2 Composants et architecture dynamique

Les deux applications sont découpés en composant actifs appelés ici threads et échangent ressources et messages. Voici les principaux threads mis en oeuvre

- **Application MPL-Learn :**

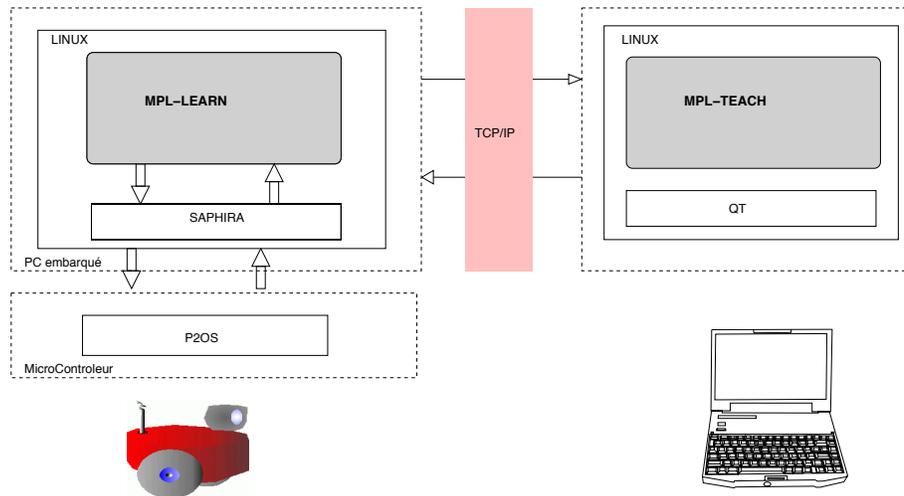


FIG. 14.9: Environnement logiciel des deux applications : *MPL-Teach* côté robot et *MPL-Learn* côté tuteur.

- **Thread Recorder** : chargé de l’enregistrement des démonstrations.
- **Thread Perception** : chargé des activités d’acquisition et de perception.
- **Thread Behavior** : chargé de la réalisation du comportement courant (algorithme *realize*).
- **Thread Control** : chargé de la communication avec l’extérieur et du contrôle global des activités.
- **Thread Interface** : chargé de l’adaptation à l’environnement logiciel (*Saphira*).
- **Application MPL-Teach** :
  - **Thread Tutor Vision** : chargé des différents feed-back à destination du tuteur, en particulier de la vision du robot.
  - **Thread Tutor Control** : chargé de la prise en compte des commandes du tuteur.
  - **Thread Synthetizer** : chargé de la phase offline d’apprentissage (alg. *generate\_pop* et *inform\_pop*).

Dans le composant *Behavior*, la boucle perception/action s’exécute à une fréquence de  $3\text{hz}$  en réalisation autonome comme durant l’enregistrement. Pour limiter la charge réseau et la charge temps réel de *MPL-Learn* le logiciel *MPL-Teach*, obtient par polling les données de feedback (vision courante, état, effecteurs) à une fréquence de  $1\text{hz}$ .

La figure 14.10 représente les principaux échanges intervenant entre les threads lors de l’en-

enregistrement des démonstrations et lors de la réalisation autonome d'un comportement.

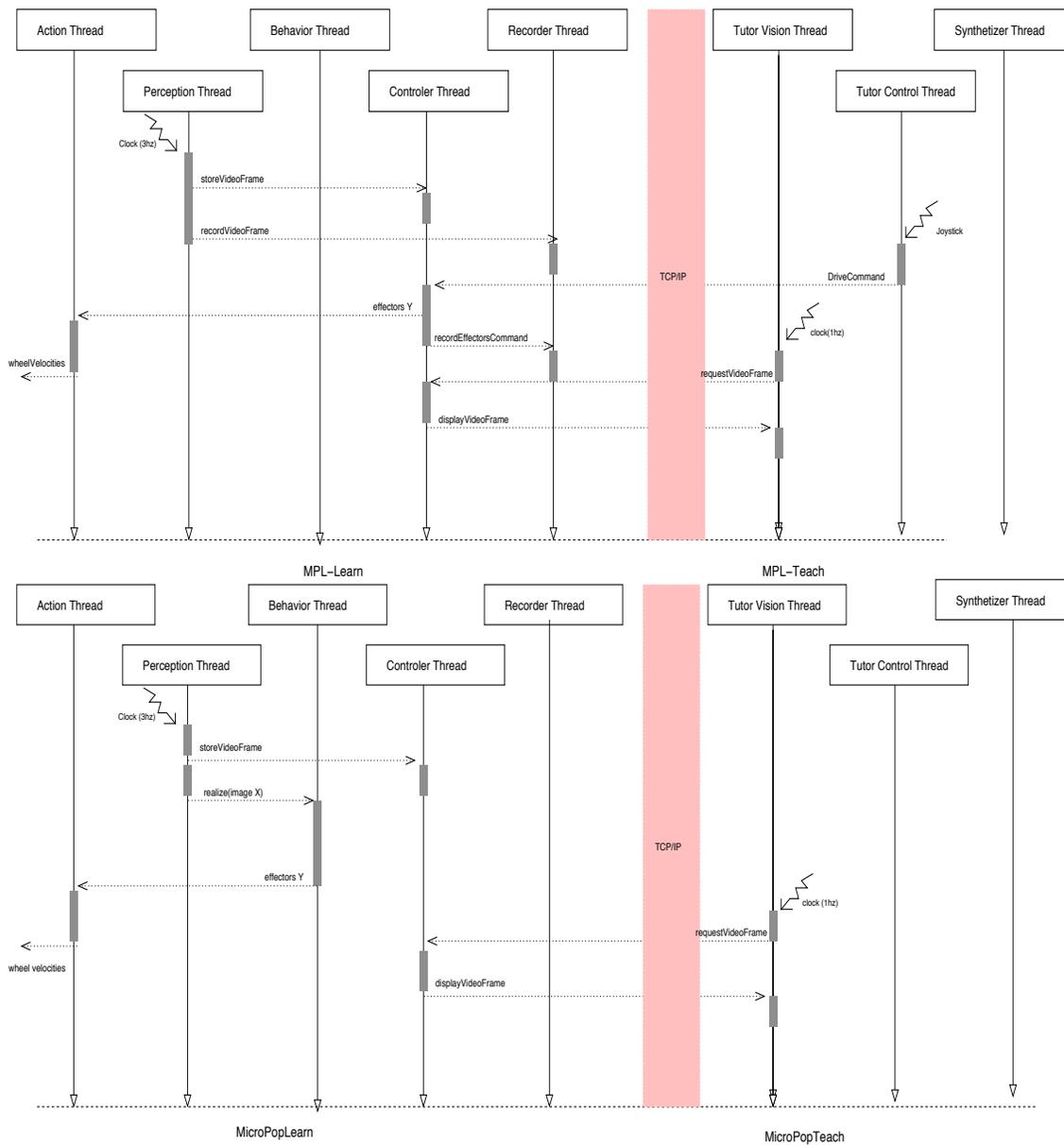


FIG. 14.10: En haut : scénario des principaux échanges en phase de démonstration. En bas : scénario des principaux échanges en phase de réalisation autonome. Les deux figures représentent par des flèches les principaux messages/événements asynchrones survenant entre les threads.

# Chapitre 15

## Acquisition de comportements

### 15.1 Introduction

Ce chapitre décrit les expérimentations d'acquisition de comportements que j'ai réalisées en environnement réel en utilisant de la plate-forme décrite dans le chapitre 14. Ces expérimentations ont toutes été effectuées dans les locaux du laboratoire LIP6 de l'université Pierre et Marie Curie, plus précisément dans les parties de l'environnement les plus typiques de locaux bureaux classiques telles que les couloirs, les endroits proches des ascenseurs ou des machines à café. Ces zones ont été utilisées sans préparation particulière. Je détaille ici la liste d'expériences suivante pour mettre en évidence les possibilités du modèle ainsi que ses limites :

- Expérience GOSTOP
- Expérience SLALOM
- Expérience DOCKING
- Expérience FUSION
- Expérience OCCLUSION

### 15.2 Expérience GOSTOP

Cette expérience a permis d'étudier les points suivants :

- L'organisation du champ perceptuel.
- La faisabilité de l'apprentissage.

Cette expérience a été conçue pour comparer plusieurs *configuration* différentes du champ perceptuel et déterminer celles qui sont adaptées à l'apprentissage de comportement dans le type d'environnement visé. Une configuration est une organisation spécifique du champ perceptuel comportant des plans et des filtres particuliers. Les différentes configurations du champ perceptuel font intervenir des filtres (Norm, Smooth) et des plans perceptuels (COLOR, SHAPE,EDGE) déjà évoqués dans le chapitre 13. Pour effectuer cette comparaison, peu de critères objectifs sont disponibles (cf. chapitre 4). Celui utilisé ici est la moyenne du carré de l'erreur (Mean Square Error ou MSE) entre la valeur enregistrée et la valeur proposée par le comportement après apprentissage. Cette valeur moyenne est calculée conformément à l'équation 15.1 sur toutes les frames  $\{X_i, Y_i\}$  d'un ensemble de démonstrations.

$$MSE(D_B) = \frac{1}{|D_B|} \sum_{D_B} |Y_i - realize_{D_B}(X_i)|^2 \quad (15.1)$$

On compare ici différentes configurations du champ perceptuel en fonction de la valeur MSE qu'elles permettent d'obtenir. Cependant celle-ci ne peut évidemment que servir de guide et ne permet pas à elle seule de dire si un comportement correspond à l'intention du tuteur.

### 15.2.1 Démonstrations

La tâche que doit accomplir le robot consiste à entrer dans une pièce et à s'approcher d'un panneau STOP. La figure 15.1 montre des images extraites d'une des démonstrations. Deux ensembles distincts de démonstrations ont été enregistrés : un ensemble d'apprentissage LS (Learning Set) et un ensemble de validation VS (Validation Set). L'ensemble LS comporte 8 démonstrations et l'ensemble VS en comporte 4. Les démonstrations de l'ensemble VS sont similaires à celle de LS et constituent d'autres exemples types du comportement attendu par le tuteur. Le comportement est appris à partir de l'ensemble LS, puis la valeur MSE est calculée sur les deux ensembles VS et LS.

### 15.2.2 Résultats

Le tableau 15.1 donne la valeur de MSE sur VS et LS pour 20 configurations différentes du champ perceptuel. L'erreur est plus faible pour l'ensemble LS que pour VS (témoignant d'un sur-apprentissage sur LS). Les cinq configurations 1, 4, 5, 6 et 9 (en gras dans le tableau)

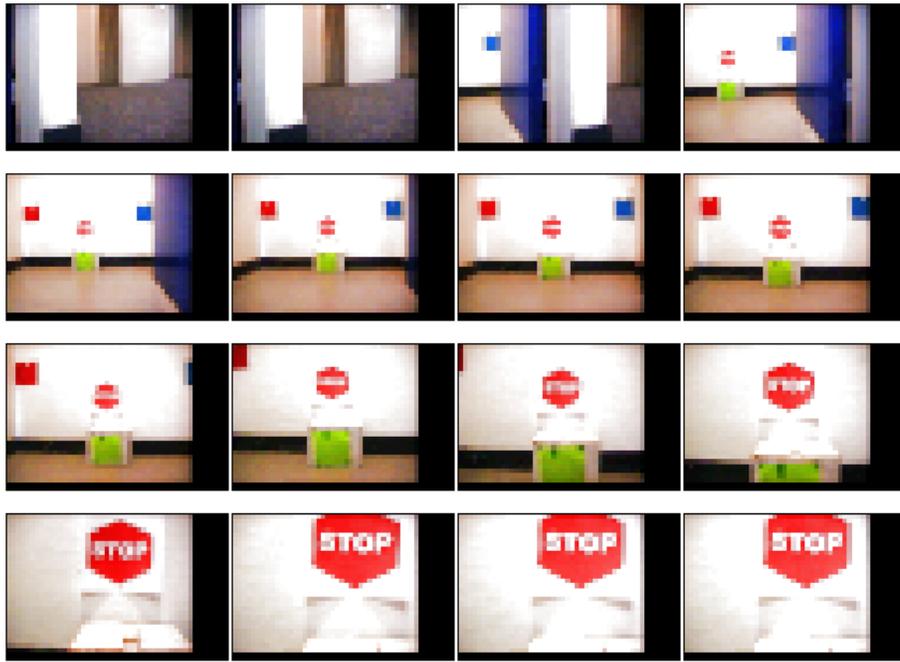


FIG. 15.1: Une des huit démonstrations de l'ensemble d'apprentissage LS (résumée)

se montrent légèrement meilleures que les autres candidats. On peut faire les remarques suivantes : un codage de la couleur prenant en compte la teinte (hsv431) semble préférable à un codage en niveau de gris (lumin32).

### 15.2.3 Configuration

Parmi les 5 configurations ( 1, 4, 5, 6 et 9) celles qui s'avèrent capable de restituer le comportement GOSTOP en situation sont celles qui ont donné de bons résultats sur VS et LS soit la configuration 1 et la configuration 5. C'est la configuration de la table 15.2 qui sera reprise pour toutes les expériences suivantes.

Aucun mécanisme d'évitement d'obstacle n'est utilisé, les sonars et bumpers équipant le robot ne sont pas pris en compte.

Config	Norm	Smooth	COLOR	SHAPE	EDGE	MSE VS	MSE LS
1	norm2	<i>rien</i>	hsv431	shapel	<i>rien</i>	<b>0.017</b>	0.00029
2	<i>rien</i>	smooth1	hsv431	shapel	<i>rien</i>	0.035	0.00073
3	<i>rien</i>	<i>rien</i>	hsv431	shapel	<i>rien</i>	0.032	0.00059
4	<i>rien</i>	<i>rien</i>	hsv431	<i>rien</i>	<i>rien</i>	<b>0.016</b>	0.00357
5	norm2	<i>rien</i>	hsv431	shapel	edge1	<b>0.017</b>	0.00021
6	norm2	<i>rien</i>	hsv431	<i>rien</i>	edge1	<b>0.014</b>	0.00194
7	norm1	<i>rien</i>	hsv431	shapel	<i>rien</i>	0.030	0.00059
8	norm3	<i>rien</i>	hsv431	shapel	<i>rien</i>	0.031	0.00060
9	norm2	smooth1	hsv431	shapel	<i>rien</i>	<b>0.014</b>	0.00054
10	norm2	smooth2	hsv431	shapel	<i>rien</i>	0.027	0.00062
11	norm2	<i>rien</i>	lumin32	shapel	<i>rien</i>	0.035	0.00011
12	<i>rien</i>	smooth1	lumin32	shapel	<i>rien</i>	0.030	0.00032
13	<i>rien</i>	<i>rien</i>	lumin32	shapel	<i>rien</i>	0.026	0.00021
14	<i>rien</i>	<i>rien</i>	lumin32	<i>rien</i>	<i>rien</i>	0.022	0.0009
15	norm2	<i>rien</i>	lumin32	shapel	edge1	0.040	0.00009
16	norm2	<i>rien</i>	lumin32	<i>rien</i>	edge1	0.023	0.00040
17	norm1	<i>rien</i>	lumin32	shapel	<i>rien</i>	0.031	0.00011
18	norm3	<i>rien</i>	lumin32	shapel	<i>rien</i>	0.032	0.00022
19	norm2	smooth1	lumin32	shapel	<i>rien</i>	0.031	0.00017
20	norm2	smooth2	lumin32	shapel	<i>rien</i>	0.040	0.00020

TAB. 15.1: MSE calculé pour différentes configurations de filtres et classeurs

<b>Configuration expérimentale</b>	
	<b>MPL-s</b> avec $\alpha = 0.20$ et mécanisme de perception active
Filtres	type Normalisation Norm2
Plan 1	Couleur, classeur hsv431
Plan 2	Forme, codage shapel

TAB. 15.2: Configuration reprise pour toute les expériences.

## 15.3 Expérience SLALOM

Cette expérience a permis d'étudier les points suivants :

- La prise en compte de situations perceptuelles variés.
- L'apprentissage à partir de peu d'exemples.
- La comparaison avec un réseau de neurones.

Dans l'expérience *SLALOM*, la tâche à apprendre consiste à effectuer un slalom entre trois points matérialisés par des boites colorées puis à s'approcher d'une boîte 'Goal' et à s'arrêter. L'environnement pour l'apprentissage de la tâche est celui représenté 15.2.



FIG. 15.2: Environnement de la tache *SLALOM*.

### 15.3.1 Démonstrations

Cinq démonstrations ont été enregistrées pour apprendre la tâche *SLALOM*. Elles ont toutes un même point de départ se trouvant en face du premier plot. Les cinq démonstrations sont résumées dans la figure 15.3. Dans une vue plus détaillée de la première démonstration (fig. 15.4) on constate que le robot ne voit que rarement les plots colorés. Il doit donc déterminer ses actions en fonction d'autres éléments de l'environnement.

La figure 15.5 montre le relevé odométrique correspondant aux cinq démonstrations. Ce relevé donne la position relative ( $x_{pos}, y_{pos}$ ) du robot depuis son point de départ d'après les valeurs de l'odomètre (cf 14.2.1.0.0). Ce type d'information est connu pour accumuler très rapidement les erreurs et n'est pas utilisé pour l'apprentissage. L'une des trajectoires montre d'ailleurs un important décalage dû à un dérapage au démarrage. Ces relevés permettent

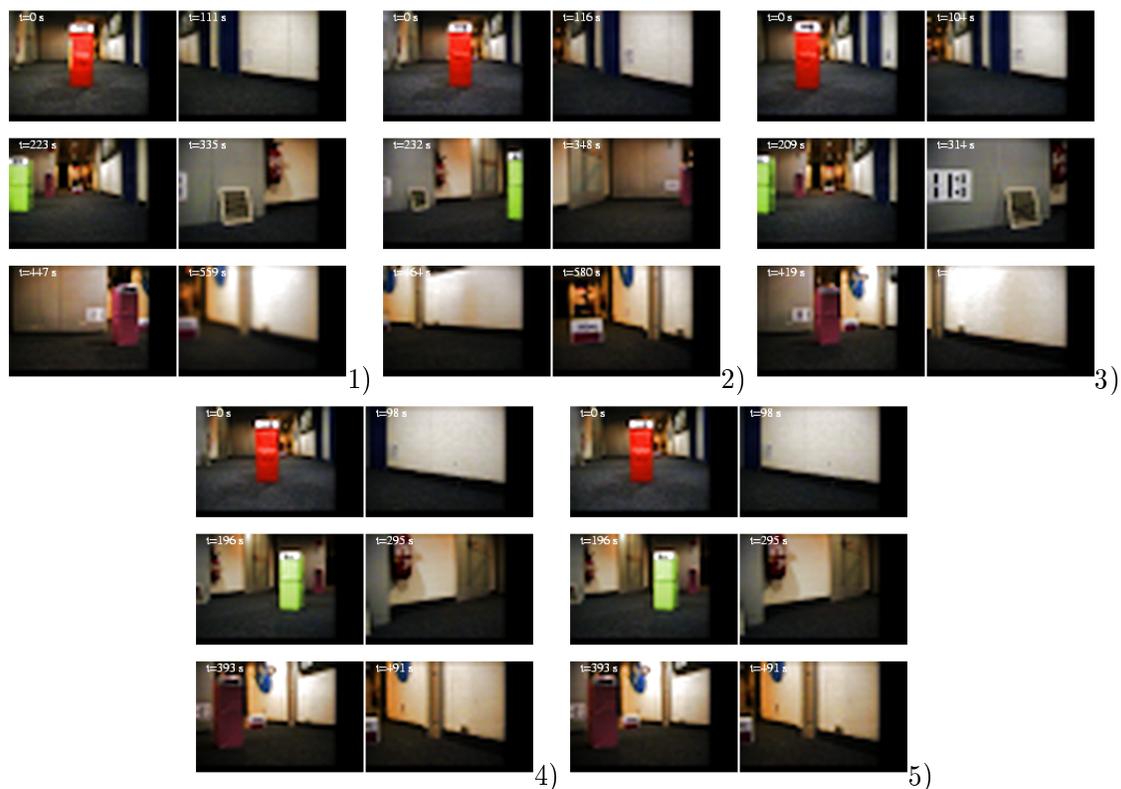


FIG. 15.3: Démonstrations utilisées pour l'expérience Slalom (résumée)

cependant ici de représenter le profil des trajectoires.

### 15.3.2 Résultats

L'apprentissage prend 2 minutes 42 secondes sur un PC Intel 366Mhz. La durée de la phase d'apprentissage est quasiment linéaire en fonction du nombre de frames pour les grandeurs considérées (fig 15.6).

La population de cellules permet d'encoder les exemples avec précision. La figure 15.7 permet de confronter pour chaque frame  $\{X_i, Y_i\}$  d'une démonstration la réponse désirée enregistrée  $Y_i$  et la réponse obtenue en stimulant la population de cellules avec les mêmes données visuelles  $X_i$ . Pour simplifier, les tracés ne représente que la variable  $Y_{left}$ .

La phase d'apprentissage génère une population de 54233 cellules dont les valeurs  $y_i$  se répartissent dans l'espace des actions comme indiqué dans la figure 15.8.

Lors de la *réalisation autonome* du comportement un succès est mesuré comme suit : le

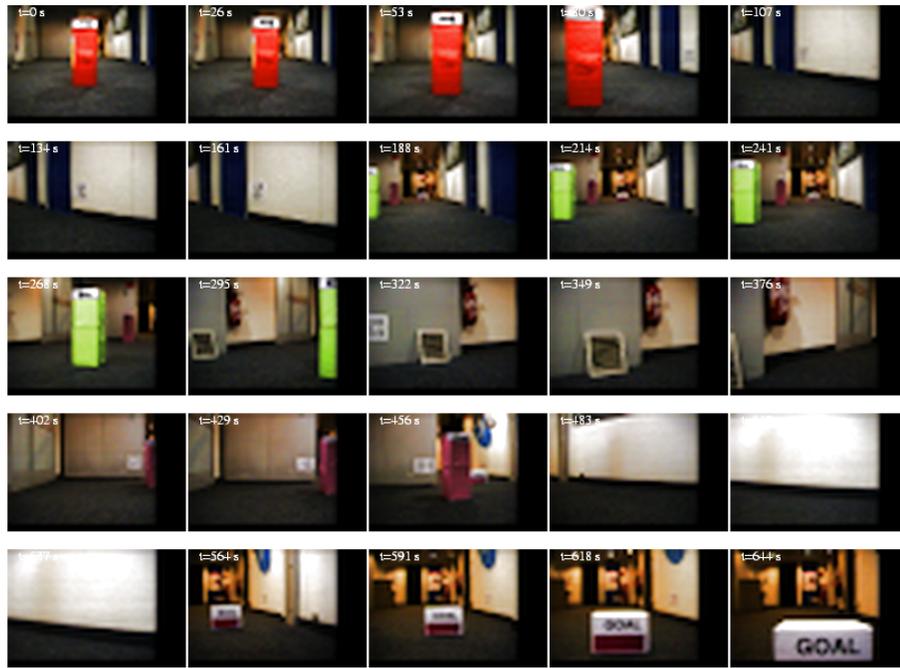


FIG. 15.4: Première démonstration (résumée)

Exemple	durée	nombre de frames
1	90s	290
2	93 s	301
3	84s	271
4	78s	255
5	80s	258
total	425s	1375

TAB. 15.3: Durée des démonstrations

robot doit avoir suivi la trajectoire générale sans toucher les plots et doit s'approcher de la boîte cible à moins de 20cm. La table 15.4 récapitule les résultats obtenus sur une session de 30 essais successifs. La tâche est découpée en trois tiers : passage du 1er plot, du second plot, du troisième et fin.

Le comportement est réactif et opportuniste, si le robot est déplacé et posé quelque part dans l'environnement il reprend le comportement à partir du point correspondant. Le mécanisme de perception active est mis en marche automatiquement en moyenne 3 fois durant une réalisation du comportement, lorsque le robot diverge trop des trajectoires connues. Le fait que des personnes passent à une allure normale dans la zone ne pose pas de problème particulier au robot. Si, par contre quelqu'un s'arrête et oblitère trop le champ visuel (occlusion à plus de 25%), le robot entre en mode perception active jusqu'à ce que la situation s'améliore.

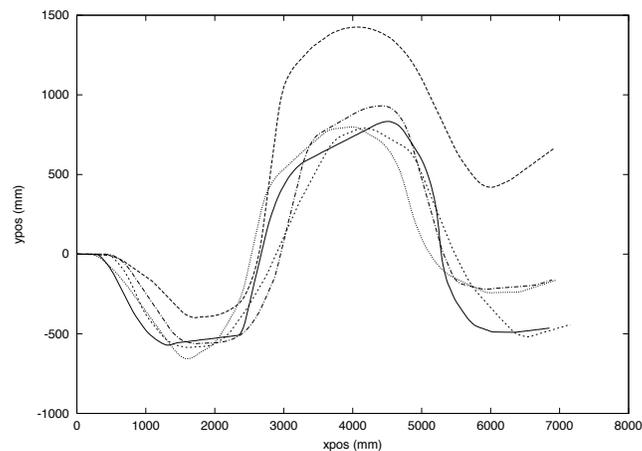


FIG. 15.5: Relevé odométrique pour les cinq démonstrations

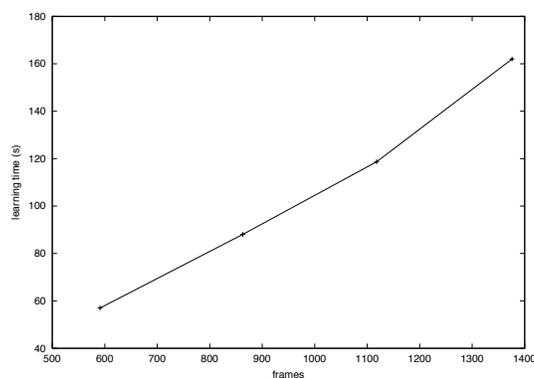


FIG. 15.6: Durée d'apprentissage en fonction du nombre de frames contenues dans les démonstrations.

Les défauts de réalisation du comportement surviennent particulièrement avant le dernier plot. A cet endroit, le robot est surtout confronté à un mur blanc devant lequel il doit d'abord s'avancer puis tourner. Il a dans ce contexte peu d'informations pour dissocier les deux actions. On verra dans le chapitre 17 que les résultats peuvent être améliorés grâce à l'apprentissage interactif.

### 15.3.3 Comparaison avec un réseau de neurones PMC

A partir des mêmes démonstrations, j'ai testé les performances d'un réseau de neurones. Dans ce cas la fonction  $Y_i = realize(X_i)$  n'est plus calculée par la population de cellules mais par un perceptron multicouches entraîné avec les couples  $\{X_i, Y_i\}$  provenant des démonstrations. Pour l'entraînement du réseau, j'ai utilisé le logiciel SNNS (Stuttgart Neural Network Simulator) de l'université de Stuttgart ([Zell *et al.*, 1992]). Les conditions d'entraînement sont les suivantes : entraînement par rétropropagation, fonction d'activation logistique, réseau com-

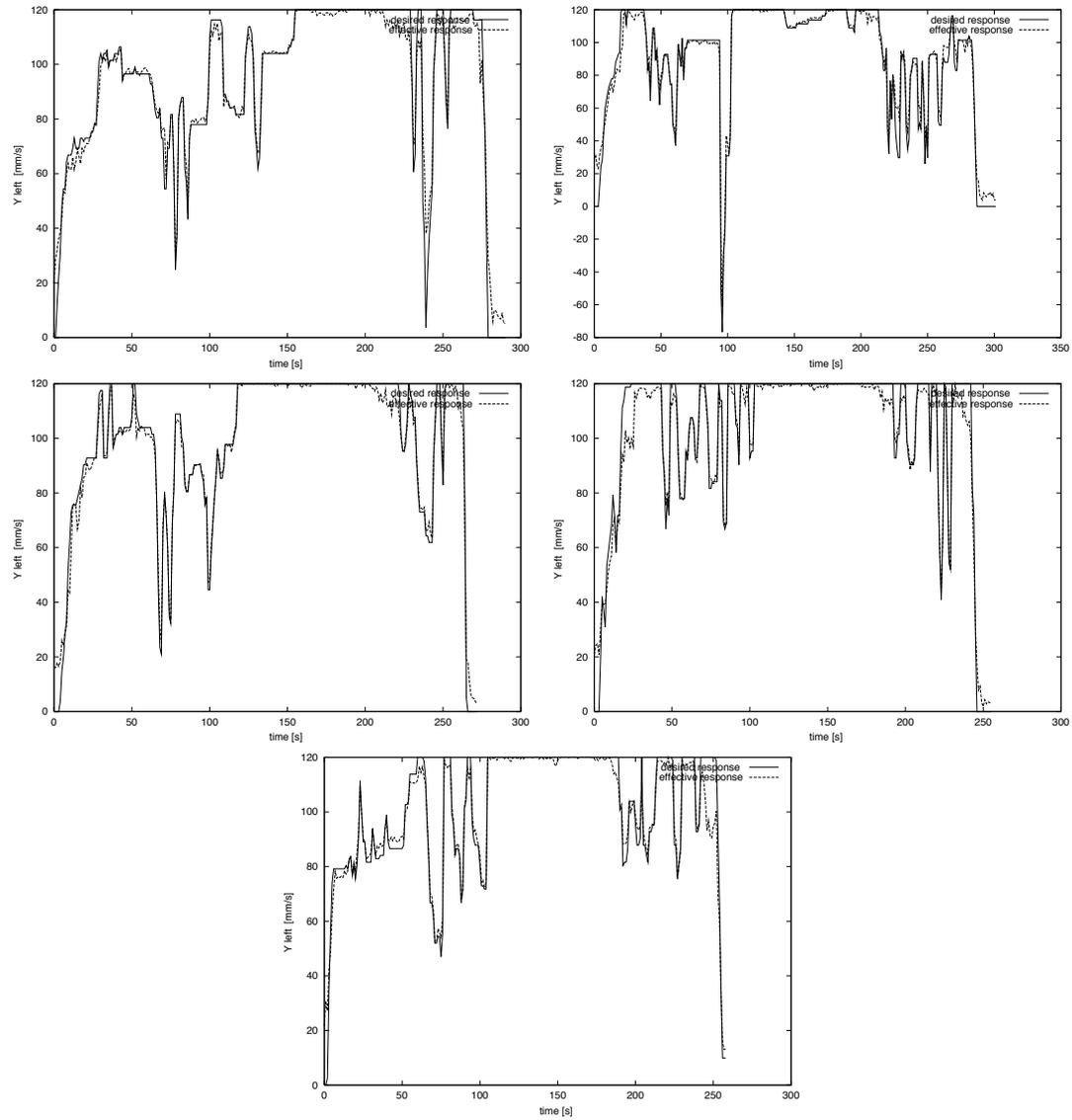


FIG. 15.7: Réponse attendue et réponse désirée pour les cinq démonstrations de la tâche SLALOM ( $Y_{left}$  uniquement)

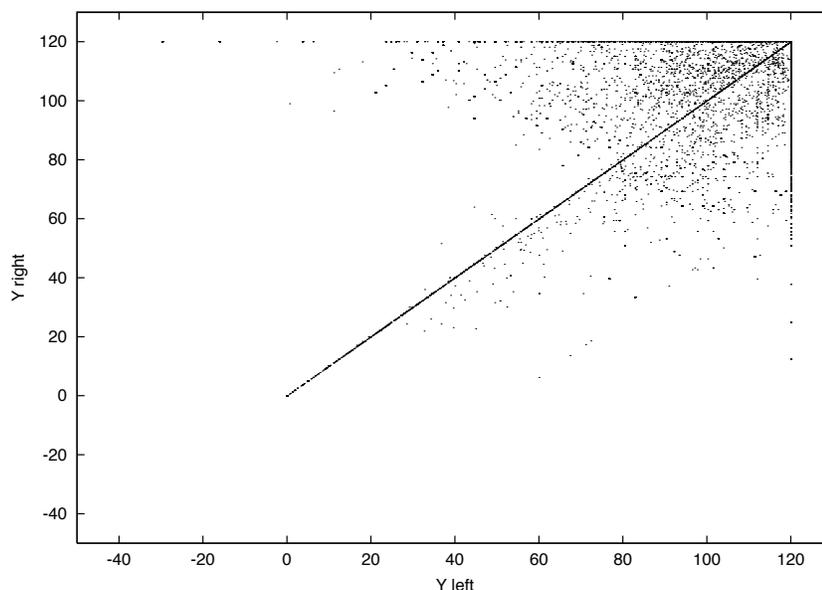


FIG. 15.8: Représentation des actions mémorisée par les 54233 cellules de  $C_{SLALOM}$ , un point dans l'espace des actions correspond à une cellule.

	Succès sur 30 essais
1 tiers de la tâche	90%
2 tiers de la tâche	83%
tâche complète	53%

TAB. 15.4: Tâche SLALOM : résultats sur 30 essais.

plètement connecté.

La table 15.3.3 donne les résultats pour différentes configurations et différentes durées d'entraînement. La configuration NN-1200-5-2 correspond à 1200 neurones en entrée (correspondant au 40x30 pixels de l'image  $X_i$ ), 5 neurones sur la couche cachée et 2 neurones de sortie correspondant à  $Y_{left}$  et  $Y_{right}$ . Les tests ont été effectués en codant pour chaque pixel sa couleur sur 256 niveaux de gris (d'autres tests reprenant l'information de couleur ont été aussi effectués sans donner de meilleurs résultats).

Comme l'indique le tableau 15.3.3, le PMC entraîné ne permet jamais de réaliser le comportement. Les tests portent sur 30 essais et sur le seul premier tiers de la tâche. En comparaison, **MPL-s** permet sur ce premier tiers d'obtenir 90% de succès. Si on entraîne plus longtemps le PMC (500 cycles) on reconnaît quelques traces du comportement lors de la restitution, mais qui restent insuffisantes pour une réalisation correcte. Dans tous les cas, le temps d'ap-

Configuration	cycles (epoch)	Temps d'apprentissage	MSE LS	succès
NN-1200-10-2	100	9min10s	0.00258	0%
NN-1200-10-2	500	46min	0.00177	0%
NN-1200-5-2	100	5min20	0.00256	0%
NN-1200-20-20-2	100	21min	0.00213	0%
<b>MPL-s</b>	1	2min42s	0.00021	90%

TAB. 15.5: Comparaison avec plusieurs configurations de réseau de neurones, donnant le pourcentage de succès sur le premier tiers de la tâche SLALOM.

prentissage est beaucoup plus long que pour la population de cellules pour une MSE dix fois supérieure sur l'ensemble d'apprentissage. Par ailleurs, le réseau de neurones est toujours actif et fournit une valeur pour les effecteurs en permanence, y compris pour les situations inconnues, ce qui produit le plus souvent une marche aléatoire pouvant s'avérer dangereuse.

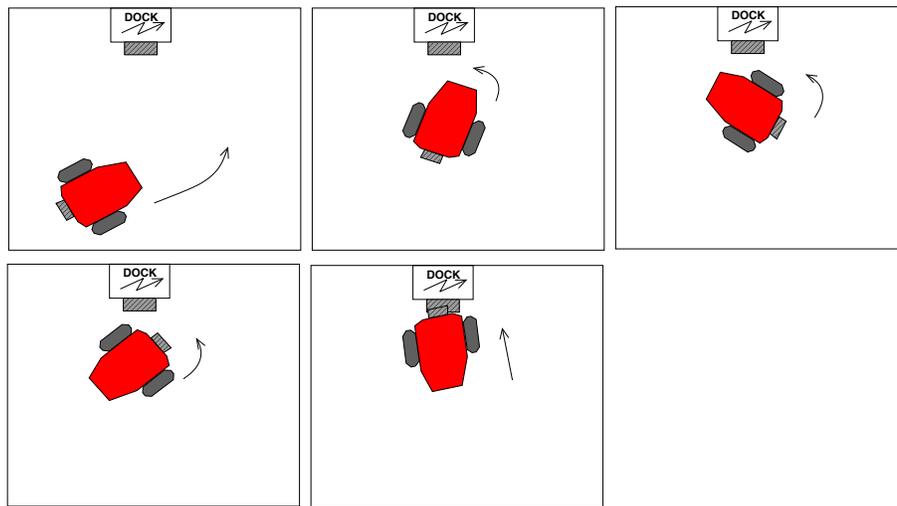


FIG. 15.9: Manoeuvre à apprendre pour faire un accostage dans le cas de la tâche docking

## 15.4 Expérience DOCKING

Cette expérience permet d'étudier les points suivants :

- L'apprentissage d'une manoeuvre complexe.
- La synthèse à partir de plusieurs variantes d'un comportement.
- Apprendre un prototype de comportement utilisable dans une application réelle.

Dans l'expérience DOCKING, la tâche à apprendre consiste à effectuer un accostage à une station de recharge. Cette tâche est typique des composants comportementaux utiles pouvant faire partie d'une application robotique réelle. Pour se recharger, le robot est équipé à l'arrière d'un becquet permettant le contact électrique et il doit s'enficher dans la station de recharge (cf. description dans la section 1.2.1).

L'objectif ici est donc d'apprendre une série de manoeuvres comme celles décrites dans la figure 15.9 seulement à partir de la vision . Celles ci doivent pouvoir être enclenchées à partir de plusieurs positions de départ situées autour de la station de recharge. Pour l'apprentissage, la station de recharge est installée dans un des couloirs du laboratoire.

Ce comportement est très difficile à programmer. Pour le besoin d'autres expérimentations dans le cadre de *MICRobES* il a d'ailleurs été programmé par François Sempé et Pierre Emmanuel Viel [Sempe *et al.*, 2002]. Il a fallu pour cela mettre au point un système de détection

de balises visuelles extrêmement précis permettant de connaître l'angle et la distance d'une balise; puis, après avoir équipé la station de recharge de la balise appropriée, intégrer cette détection dans un comportement programmé.

### 15.4.1 Démonstrations

Pour l'apprentissage de ce comportement, 8 démonstrations ont été enregistrées. La première est détaillée dans la figure 15.10. On y remarque que le robot ne voit pas en permanence la station. Les huit démonstrations sont résumées dans la figure 15.12 et correspondent aux positions de départ repérées dans la figure 15.11.

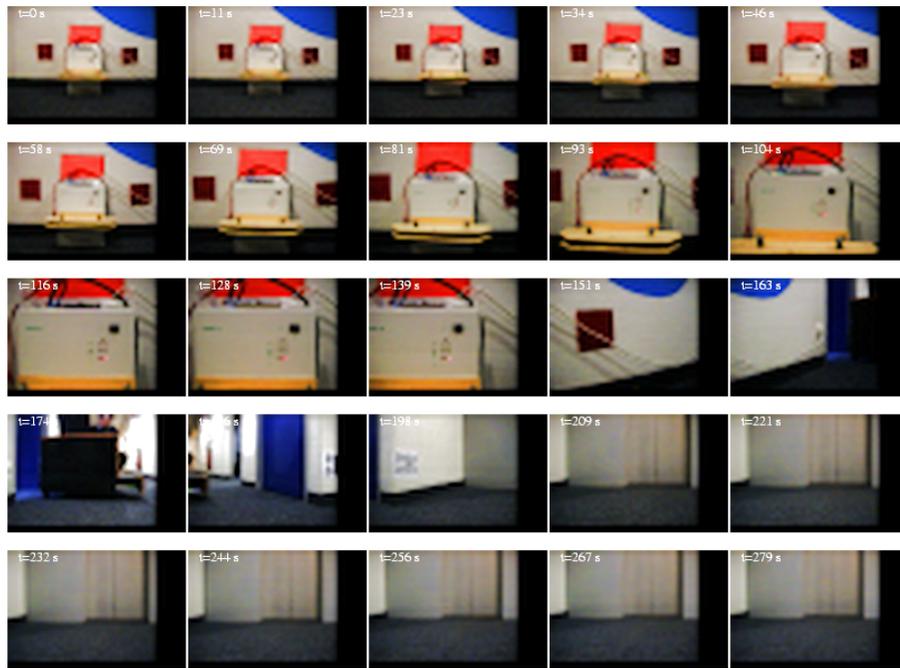


FIG. 15.10: Expérience DOCKING : images successives extraites de la première démonstration.

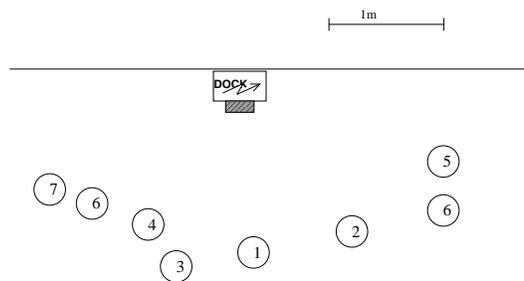


FIG. 15.11: Expérience DOCKING : Position de départ des huit démonstrations.

### 15.4.2 Résultats

Contrairement à la tâche SLALOM où les démonstrations démarraient toutes du même point, le mécanisme de synthèse doit ici fusionner dans un même comportement des variantes sensiblement différentes. Lors de la *réalisation autonome* du comportement, les positions de départ pour le comportement sont distribuées autour de la station. Le mécanisme de perception active est systématiquement activé lors de ces réalisations. Sur 30 essais le robot parvient à réaliser parfaitement le comportement (jusqu'à l'enfichage correct du becquet matérialisé par l'allumage d'un témoin de charge sur la station) pour huit cas. Les cas où le robot se positionne correctement, mais heurte la station en fin réalisation, doivent aussi être considérés comme des succès ( un becquet plus adapté, comportant un accrochage magnétique par exemple, permettrait de remédier à ce problème). En intégrant les deux cas de figure le pourcentage de succès est de 46%. Les cas d'échec se produisent essentiellement lorsque le robot démarre trop près du mur et qu'il n'a pas la possibilité de se placer correctement face à la station.

	Succès sur 30 essais
Jusqu'à l'allumage du témoin de charge	27%
Jusqu'au positionnement correct	46%

TAB. 15.6: Tâche DOCKING : résultats sur 30 essais.

Le robot n'utilise ici aucun mécanisme d'évitement d'obstacle et doit donc se positionner grâce à sa seule vision. La combinaison (par subsomption) d'un mécanisme d'évitement d'obstacles utilisant les sonars avec le comportement d'accostage permettrait sans doute, en conjonction avec les modifications matérielles, d'améliorer considérablement le comportement final.

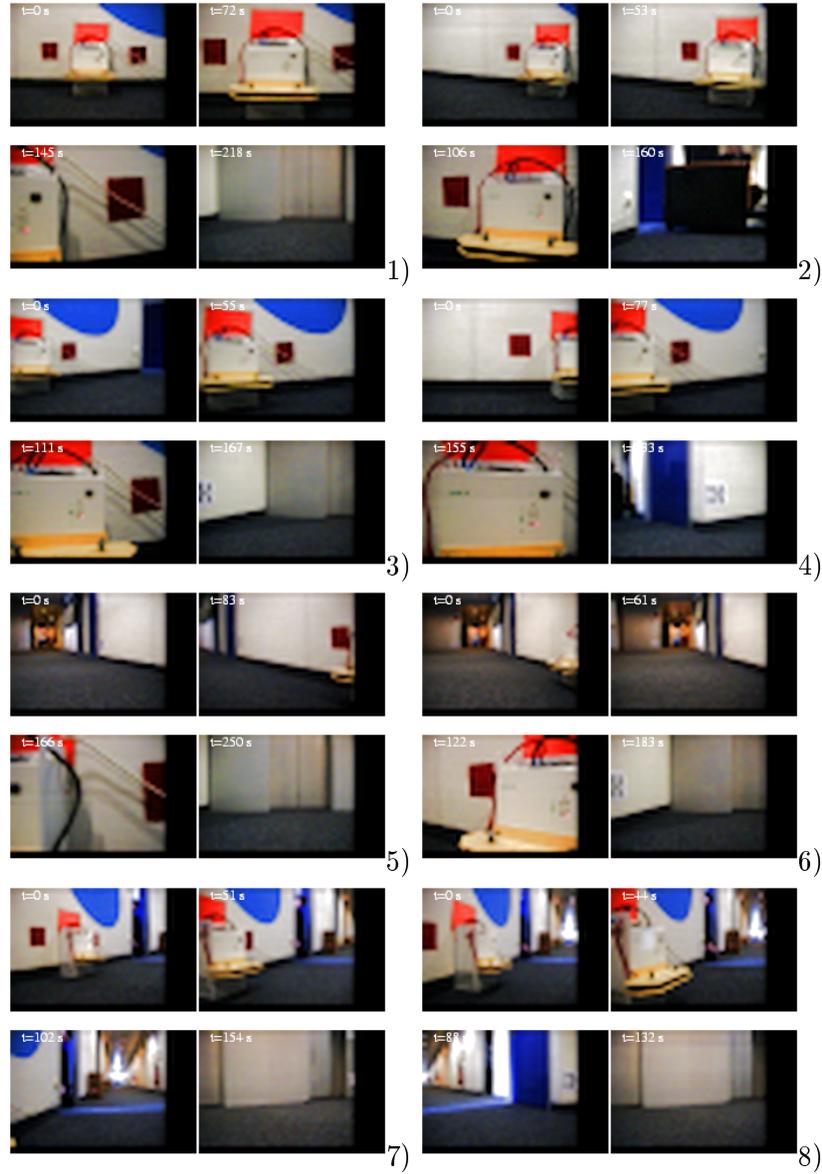


FIG. 15.12: Expérience DOCKING : les huit démonstrations résumées par quatre images.

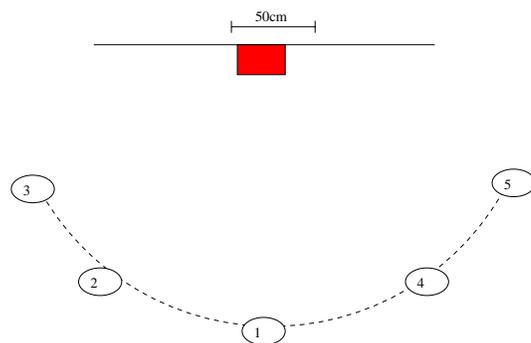


FIG. 15.13: Expérience FUSION, Points de départ des cinq démonstrations pour un même contexte.

## 15.5 Expérience FUSION

Cette expérience a permis d'étudier les points suivants :

- La fusion de comportements acquis dans des contextes différents.
- Le degré d'abstraction d'un comportement appris.

Dans l'expérience FUSION, le comportement à apprendre est très simple : il consiste à s'approcher d'une boîte de couleur et à s'arrêter. Dans cette expérience, le même comportement est appris à partir de démonstrations provenant de trois contextes différents. L'objectif est de vérifier s'il est possible de fusionner dans une même population de cellule plusieurs versions d'un comportement. Il s'agirait alors de fournir une sorte de "proto-abstraction" du comportement qui, à un même nom générique, ferait correspondre plusieurs versions ancrées dans des environnements différents.

### 15.5.1 Démonstrations

Les démonstrations sont obtenues dans trois contextes différents : A, B et C. Dans chacun de ces contextes, cinq démonstrations ont été enregistrées autour d'une boîte orange à partir des positions de la figure 15.13. Les figures 15.14, 15.15 et 15.16 correspondent aux trois contextes et montrent les images de départ de chacune des  $3 \times 5$  démonstrations formant l'ensemble  $D_{FUSION-ABC}$ .

### 15.5.2 Résultats

On considère quatre versions du comportement : FUSION-A, FUSION-B, FUSION-C et FUSION-ABC. Les trois premières correspondent aux comportements appris respectivement

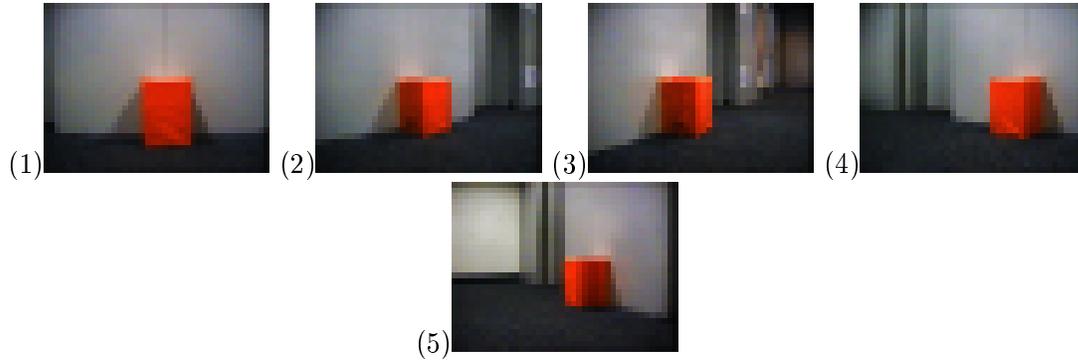


FIG. 15.14: CONTEXTE A : Première image de chacune des 5 démonstrations enregistrées dans le contexte A

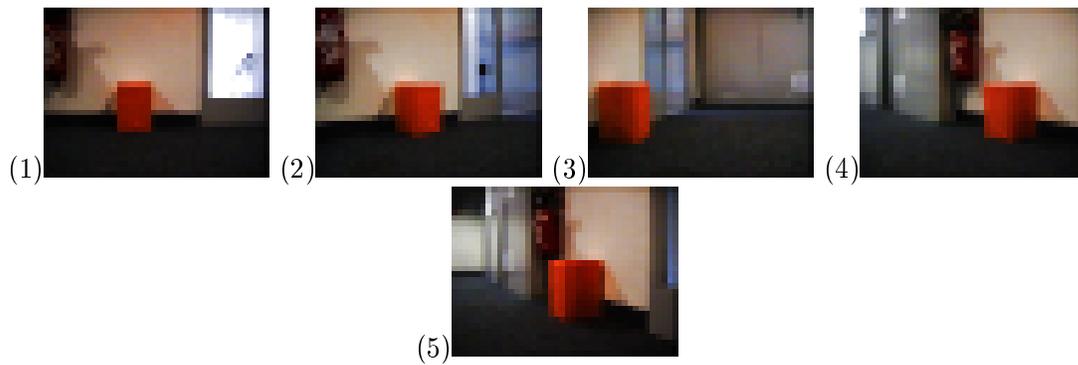


FIG. 15.15: CONTEXTE B : Première image de chacune des 5 démonstrations enregistrées dans le contexte B

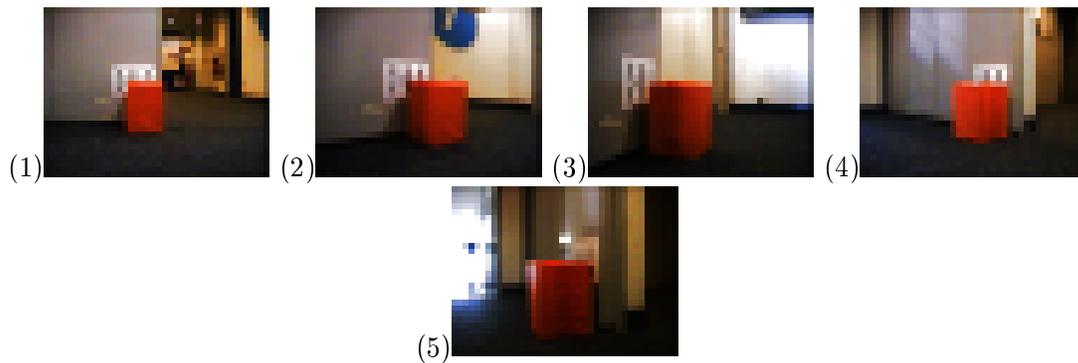


FIG. 15.16: CONTEXTE C : Première image de chacun des 5 démonstrations enregistrées dans le contexte C

à partir des démonstrations des contextes A, B, et C. FUSION-ABC correspond lui au comportement appris à partir des quinze démonstrations. Par ailleurs on considère un quatrième contexte D (la boîte orange posée devant un mur blanc) d'où aucune démonstration n'est issue.

	succès/tentative
FUSION-A dans contexte A	8/10
FUSION-B dans contexte B	8/10
FUSION-C dans contexte C	7/10
FUSION-ABC dans contexte A	8/10
FUSION-ABC dans contexte B	7/10
FUSION-ABC dans contexte C	8/10
FUSION-ABC dans contexte inconnu D	0/10

TAB. 15.7: *Expérience FUSION*

Le comportement est essayé à partir de positions aléatoires sur l'arc de cercle de la figure 15.13. La qualité de restitution du comportement FUSION-ABC est sensiblement égale à celle des comportements FUSION-A, FUSION-B et FUSION-C pris séparément. On dispose par le biais de la fusion d'un comportement correspondant à trois environnements différents. Le procédé de fusion peut aussi être utilisé pour apprendre un comportement dans des situations différentes, la nuit et le jour par exemple. Cependant, si on place la boîte orange dans un quatrième environnement D, inconnu du robot, celui-ci reste immobile car trop peu de cellules sont actives. Les micro-percepts des cellules sont répartis de manière uniforme sur la boîte et le fond, et la présence de la boîte seule ne suffit pas à activer suffisamment de cellules. Le comportement FUSION-ABC ne semble pas correspondre au comportement abstrait "s'approcher de la boîte orange".

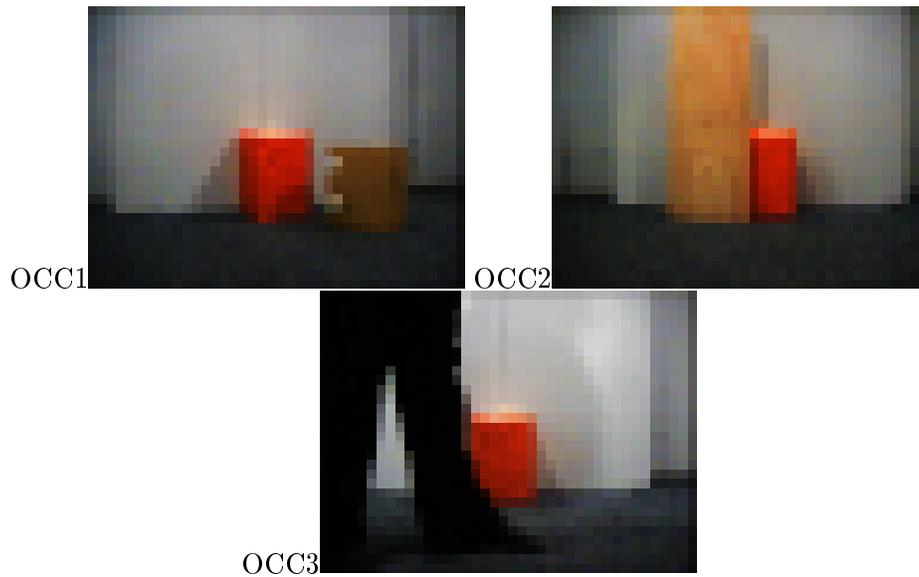


FIG. 15.17: Trois conditions d'occlusion. Le champ visuel est occulté par un objet de taille moyenne dans (OCC1) et de grand taille dans (OCC2); dans (OCC3) une personne passe régulièrement devant le robot.

## 15.6 Expérience OCCLUSION

Cette expérience à permis d'étudier les points suivants :

- Robustesse aux occlusions statiques.
- Robustesse aux occlusions dynamiques.

Dans cette expérience on reprend les démonstrations de l'expérience FUSION-ABC et le comportement appris dans les trois contextes. Le comportement est ensuite réalisé dans différentes conditions d'occlusion représentées dans la figure 15.17.

- OCC1 Occlusion statique : Un objet de moyenne taille masque une partie du champ visuel en permanence.
- OCC2 Occlusion statique : Un objet de grand taille masque une partie du champ visuel en permanence.
- OCC3 Occlusion dynamique : quelqu'un passe devant le robot toutes les 2 secondes à une vitesse de marche normale.

### 15.6.1 Résultats

Dans la situation d'occlusion OCC1 le robot reproduit le comportement sans être gêné ; La présence de l'objet supprime des cellules qui auraient normalement été activées, cependant le nombre de cellules activées reste suffisamment important pour guider le robot. On retrouve les même chiffres que dans la section 15.5 pour le comportement FUSION-ABC dans l'environnement A. Dans les cas où l'objet se trouve sur la trajectoire du robot, celui-ci le pousse.

Dans la situation d'occlusion OCC2 le robot reproduit le démarrage du comportement sans être gêné mais dès qu'il arrive près de la boîte orange la situation perceptuelle est trop différente de celles vues au cours des démonstrations : le robot ne peut terminer le comportement et est en permanence en perception active. La qualité de restitution du comportement dépend directement de la proportion du champ visuel qui est occulté. Par ailleurs dès que la condition d'occlusion est levée (retrait du panneau) le robot repart normalement.

Dans la situation d'occlusion OCC3 le passage d'une personne ne gêne jamais le robot. Ce passage se produit suffisamment rapidement pour que le robot retrouve vite une situation connue. On retrouve les même chiffres que dans la section 15.5 pour le comportement FUSION-ABC dans l'environnement A.

## 15.7 Conclusion

Les expériences décrites dans les précédents paragraphes démontrent *la faisabilité* de l'apprentissage de comportements avec les moyens mis en oeuvre : données issues d'interaction avec l'environnement, population de cellules, percepts élémentaires.

Ces expériences ont en particulier montré que **MPL** permet de :

- Apprendre des comportements à partir de quelques démonstrations.
- Apprendre rapidement et dans l'environnement réel.
- Apprendre sans fournir d'information spécifique sur les tâches.
- Prendre en compte des situations perceptuelles variées.
- Apprendre des comportements comportant des manoeuvres complexes typiques de comportements utiles (cf. DOCKING).

- Fusionner des comportements appris dans des contextes différents.
- Obtenir des comportements robustes aux occlusions dynamiques et à certaines occlusions statiques.

D'autres expériences sont nécessaires pour continuer à qualifier le champ d'application du modèle et pour caractériser plus précisément les impacts de la structuration des plans perceptuel. Mais la nature même des expériences en robotique fait qu'elles sont coûteuses, il n'est pas possible de les multiplier comme cela serait possible dans un environnement strictement logiciel. Cette contrainte doit être intégrée dans le processus de recherche dès lors que l'on parle d'environnements et de robots réels. De plus, l'expérimentateur dispose de peu de mesures objectives pour qualifier les expériences, la localisation précise du robot n'est pas toujours connue. La conception d'un *banc de mesures transportable* fournissant une localisation précise du robot (télémètre laser) et intégrant un dispositif de surveillance qui permette de répéter de nombreuses fois les expériences, est sans doute nécessaire à terme.

Les avantages et limites du modèle et la nature du principe inductif sont discutés plus en détail dans le chapitre suivant.



# Chapitre 16

## Propriétés du modèle MPL

### 16.1 Introduction

Les expériences du chapitre 15 ont montré qu'il est possible d'apprendre des comportements d'usage de l'environnement à partir d'une plateforme expérimentale mettant en oeuvre le modèle **MPL-s**. Des comportements tels que ceux correspondant aux tâches SLALOM et DOCKING sont rapidement appris à partir de peu de démonstrations. Des défauts subsistent, dont certains peuvent sans doute être résolus par des ajustements techniques (ajout d'un mécanisme d'évitement d'obstacle dans le cas de DOCKING et d'un dispositif d'accrochage magnétique, échantillonnage modifié en présence de peu d'information dans le cas de SLALOM) alors que d'autres relèvent de limites plus structurelles du modèle. Après cette phase expérimentale, ce chapitre revient sur les propriétés attendues et discute la réponse fournie par **MPL**.

### 16.2 Retour sur les propriétés attendues

A ce stade il importe de revenir sur les propriétés attendues énoncées dans le chapitre 5 et d'examiner quelles sont les réponses du système **MPL**.

1. *EDUC : Propriétés relatives à l' Education :*

L'apprentissage s'effectue sous la conduite du tuteur humain. Celui-ci n'a pas besoin d'être un expert et doit simplement "montrer" au robot des variantes du comportement qu'il souhaite. Contrairement à beaucoup de techniques, l'apprentissage est très rapide

et permet à l'utilisateur de se faire rapidement une idée de l'impact des exemples qu'il a fournis. La représentation d'un comportement est intelligible (comparée à celle d'un réseau de neurones) et peut être envisagée comme un ensemble de micro-règles. Le tuteur ne peut cependant envisager que de transmettre des comportements concrets.

2. *ENV : Propriétés relatives à l'Environnement :*

L'apprentissage s'effectue dans l'environnement réel et n'est pas tributaire de sa modélisation. L'environnement choisi pour les expérimentations (cf. chapitre 15 est représentatif des locaux de bureaux non-préparés. La structure utilisée pour le champ perceptuel devrait certainement être adaptée pour des environnements extérieurs ou très texturés (comportant par exemple du papier peint ou des moquettes non uniformes).

3. *PERC : Propriétés relatives aux Perceptions :*

Le système détermine automatiquement les percepts nécessaires à partir des démonstrations et ne nécessite pas de connaissances à priori sur la tâche visée. La nature distribuée de la perception sur une population de cellules permet de prendre en compte le bruit issu de l'environnement et des capteurs. Le système n'est pas pour le moment très robuste aux changements francs d'illumination (en particulier en lumière du jour), ceux-ci pouvant affecter sensiblement la gammes des couleurs visibles.<sup>29</sup>

4. *INTEGR : Propriétés relatives à l'Intégration :*

Le système proposé permet de constituer un ensemble de comportements référencés chacun par un symbole. Un mécanisme de planification ou de sélection de l'action peut évaluer la valeur  $U_B(X)$  dans un contexte donné pour différents comportement et choisir de les déclencher sur cette base et sur celle d'informations complémentaires (priorité, plan, etc.) Il peut s'intégrer dans une application robotique complète en permettant d'apprendre des comportements réutilisables. Il est notamment envisageable d'apprendre d'abord des comportements élémentaires séparés, puis, en montrant des séquence plus longues, de déterminer la suite de comportements activés en se référant aux variations de  $U_B(X)$ .

---

<sup>29</sup>Le problème de la constance de la couleur est de toutes façons un problème difficile en soi [Brainard and Freeman, 1997].

## 16.3 Limites structurelle de **MPL**

Il est envisageable de remédier aux différents défauts constatés par des adaptations portant sur la plateforme expérimentale, les plans du champ perceptuel ou la stratégie d'échantillonnage mais trois limites plus importantes sont rencontrées par **MPL**. Elles sont de nature plus structurelles que techniques et portent sur :

1. L'interaction différée.
2. Le niveau d'abstraction.
3. La structuration temporelle de l'action.

Dans le processus d'apprentissage, l'interaction entre tuteur et robot est différée et passe par des phases successives de démonstration et de réalisation. A mesure qu'un comportement converge vers l'intention du tuteur celui-ci souhaite intervenir beaucoup plus directement uniquement sur les points en défaut, et sans devoir passer par une nouvelle démonstration. Lorsqu'un comportement est proche de fonctionner correctement, l'impact d'une nouvelle démonstration pour améliorer un point de détail est beaucoup plus difficile à mesurer. Le chapitre 17 étudiera la possibilité d'utiliser **MPL** dans un cadre plus interactif en tirant parti de la représentation utilisée.

L'expérience **FUSION** montre qu'il est possible d'agréger plusieurs versions d'un même comportement dans une seule population de cellules; mais le comportement obtenu n'est pas pour autant un comportement abstrait permettant de prendre en compte des situations nouvelles. La possibilité d'apprendre un comportement concret sans modèle à priori de l'environnement se paye en contre-partie par un très faible niveau d'abstraction.

Il n'y a en fait pas d'informations dans les démonstrations qui permettent de construire les abstractions nécessaires (et qui nous paraissent évidentes) : le fait qu'une boîte de couleur soit souvent présente dans l'image n'est pas plus proéminent que le fait qu'un coin de mur gris soit lui aussi présent dans l'image. Pour construire des abstractions permettant d'étendre le champ d'application d'un comportement, il sera nécessaire que le tuteur intervienne d'une manière ou d'une autre pour montrer ce qui compte et ce qui peut être négligé.

Le modèle **MPL** ne permet pas en l'état de prendre en compte les aspects temporels d'un comportement. Celui appris dans la tâche SLALOM est uniquement réactif, puisqu'il n'utilise que les perceptions courantes  $X_t$  mais pas les perceptions passées  $X_{t-1}, X_{t-2} \dots$  pour décider de l'action à effectuer au temps  $t$ . Cela pose problème pour des comportements comportant une ambiguïté perceptuelle au sein d'une même séquence d'actions. Considérons par exemple un comportement où le robot doit s'avancer vers un objet puis s'éloigner en reculant du même objet : dans ce cas le robot devra savoir agir différemment pour des perceptions identiques (mêmes valeurs pour  $X_t$ ) selon le contexte temporel. Il lui est alors nécessaire de tenir compte du passé pour prendre la bonne décision.

Il est en général possible d'obtenir une représentation structurée d'une séquence d'événements en utilisant par exemples des chaînes de Markov : l'ambiguïté perceptuelle a été étudiée de manière théorique dans le cadre des POMDP (Partially Observable Markov Decision Process) [Kaelbling *et al.*, 1995]. Mais, compte tenu de la représentation distribuée de l'action dans **MPL** ce problème de la structuration temporelle de l'action est difficile et demeure ouvert. J'ai expérimenté des plans perceptuels intégrant la rémanence de l'action ou de la perception sous différentes formes, mais il ne m'a pas été possible, pour des questions de temps, de procéder à des expérimentations complètes qui puissent figurer dans ce mémoire.

## 16.4 Principe inductif

Le principe inductif utilisé dans le modèle d'apprentissage repose sur trois aspects :

- La combinaison statistique de décisions élémentaires.
- La compression des démonstrations.
- Un biais perceptuel structurel.

### 16.4.1 Combinaison de décisions

Les cellules sont des entités indépendantes non hiérarchisées et le mécanisme de réalisation s'apparente à un mécanisme de vote dans lequel chacune des cellules actives vote pour l'action qui lui correspond. Les méthodes qui combinent les résultats de plusieurs sous-systèmes pour obtenir un résultat final sont appelés *Méthodes d'ensembles* ou "Méthodes d'apprentissage par combinaison de décisions" [Cornuéjols and Miclet, 2002]. Comme l'indique T.G Dietterich [Dietterich, 2000], elles ont pour avantage de représenter statistiquement une décision à partir

de plusieurs points de vue différents. Dans le cas du modèle **MPL-s**, la décision qui devrait théoriquement être prise en fonction de la connaissance de  $F(Y|X, D_B)$  (cf. chapitre 8.4) est prise à partir de décisions élémentaires reposant sur la connaissance pour les percepts  $x_i$  de  $F(Y|x_i, D_B)$ .

### 16.4.2 Compression des démonstrations

Le principe inductif utilisé peut par ailleurs être vu comme une compression d'information. Le mécanisme d'échantillonnage utilisé dans la phase d'apprentissage de **MPL-s** réalise une compression des données provenant des démonstrations tout en conservant la distribution initiale des actions (fig. 16.4.2).

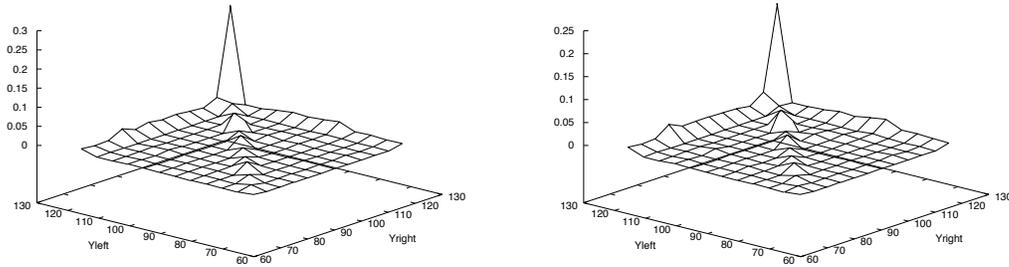


FIG. 16.1: Malgré la compression d'information effectuée lors de l'apprentissage, la distribution des actions dans les démonstrations (à gauche) et dans la population de cellules (à droite) sont identiques. Ici, pour le cas des démonstrations du comportement de la tâche SLALOM

### 16.4.3 Biais structurel

Dans **MPL-s**, le principal biais est structurel et tient à l'utilisation de conjonctions de micro-percepts. Ceux-ci permettent une généralisation à partir des exemples meilleures que celle qu'autoriseraient des macro-percepts. La figure 16.4.3 montre deux vues provenant de deux démonstrations différentes et une troisième rencontrée par le robot lors de la réalisation autonome. Lors de cette réalisation, des éléments provenant des deux démonstrations peuvent être utilisés simultanément pour prendre en compte une *situation intermédiaire inconnue* ; ainsi les éléments  $a$  et  $d$  dans la figure 16.2 peuvent être activés, traduisant ainsi l'emprunt fait à chacune des démonstrations. La fusion des percepts issus de sources distinctes est rendue possible par l'utilisation de tessels élémentaires. L'utilisation de macro-percepts (tels que des

formes géométriques) nécessiterait, quant à elle, de savoir d'abord reconnaître un même phénomène dans des démonstrations différentes, de formuler des hypothèses sur ses éventuelles déformations et de le retrouver dans les perceptions courantes.

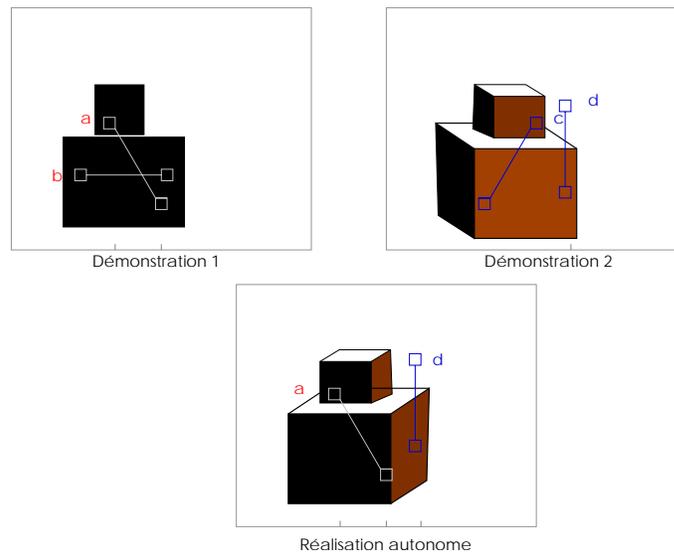


FIG. 16.2: Lors de la réalisation autonome, la tessellation permet d'emprunter des éléments provenant de démonstrations différentes.

## 16.5 Conclusion

Le système proposé permet effectivement d'apprendre des comportements d'usage de l'environnement. Les différents systèmes présentés dans le chapitre 6 ne permettent pas d'obtenir les mêmes résultats, soit parce qu'ils visent finalement un autre objectif (ie. l'imitation), soit parce qu'ils reposent sur une vision restreinte et préparée de l'environnement ; c'est le cas par exemple des travaux suivants, cités dans le chapitre 6 : [Bentivegna and Atkeson, 2000], [Pomerleau, 1993], [Friedrich and Dillman, 1995], [Billard, 1999], [Demiris and Hayes, 1996], [Hayes and Demiris, 1994], [Dorigo and Colombetti, 1998] et [Nicolescu and Mataric, 2001].

Des solutions aux limites structurelles rencontrées seront proposées dans la quatrième et dernière partie. Le chapitre 17 montrera qu'il est possible de concevoir un système d'apprentissage interactif à partir de **MPL** pour remédier au problème de l'apprentissage différé. Une solution au problème du niveau d'abstraction des comportements sera suggérée dans le chapitre 18.

Quatrième partie

**Apprentissage Interactif et  
Perspectives**



# Chapitre 17

## Vers l'apprentissage interactif

### 17.1 Introduction

Les expérimentations du chapitre 15 ont montré que le modèle **MPL** permet de transmettre au robot des comportements situés. Cependant il reste des situations dans lesquelles le comportement doit être adapté pour être plus précis et mieux correspondre à l'intention de l'utilisateur. Cette adaptation est requise en particulier lorsque le robot manque d'informations perceptuelles pour déterminer correctement ses actions. Par ailleurs, le schéma d'apprentissage par démonstrations suivi jusqu'ici permet d'intégrer l'utilisateur dans la boucle d'apprentissage mais reste décomposé en deux séries de phases distinctes : les phases de démonstrations et les phases d'essais. Or, il serait clairement préférable de permettre à l'utilisateur d'intervenir *en ligne* pour adapter rapidement un comportement et se placer dans un cadre plus direct et plus intuitif, proche de l'idée que l'on peut se faire d'un processus éducatif.

Ce chapitre propose d'intégrer le modèle **MPL-s** dans un processus plus complet qui aboutisse à un apprentissage interactif, dont la fonction principale sera de permettre d'affiner l'adaptation d'un comportement en donnant au tuteur des moyens plus immédiats pour "l'éducation" du robot. Il s'appuie, à partir du modèle **MPL**, sur l'étude expérimentale que j'ai encadrée et qui a été conduite par Clément Raievsky lors de son stage de DEA au LIP6 [Raievsky, 2002].

## 17.2 Principes

Le schéma d'apprentissage interactif que j'ai proposé et expérimenté repose sur trois principes :

1. La nécessité d'une phase d'amorçage et d'une phase d'adaptation.
2. La possibilité d'utiliser des opérateurs élémentaires pour l'adaptation du comportement.
3. L'établissement d'un processus éducatif interactif entre le tuteur et le couple robot/environnement.

### 17.2.1 Amorçage et adaptation

Il est difficile de commencer un apprentissage interactif *ab-initio* avec le robot puisque le modèle n'intègre aucune description de la tâche visée et ne manipule que des percepts élémentaires. Au démarrage le robot n'agit pas du tout ou bien aléatoirement. Pour remédier à ce problème, le cycle complet d'apprentissage comporte donc une *phase d'amorçage* qui peut ensuite être suivie d'une *phase d'adaptation* (fig. 17.1). La phase d'amorçage correspond à une phase de synthèse supervisée identique à celles déjà présentées dans le chapitre 15. On peut se représenter le comportement visé comme étant l'approximation d'une densité de probabilité conditionnelle  $F(Y|X, \mathcal{E})$  indiquant la probabilité d'effectuer une action sachant les données sensorielles courantes et l'environnement  $\mathcal{E}$ . Au départ, le robot ne sait rien hormis ce qui lui vient de quelques prédispositions structurelles et cette densité est uniforme. La phase d'amorçage permet en quelque sorte d'amener rapidement le robot dans un état utilisable en façonnant la densité  $F(Y|X, \mathcal{E})$  à partir de quelques exemples.

*Une fois cet amorçage effectué il devient possible d'adapter localement le comportement puisque l'on dispose d'un comportement partiellement prévisible.*

### 17.2.2 Opérateurs élémentaires d'adaptation

L'hypothèse faite ici est qu'il est possible d'utiliser des modalités d'adaptation simples pour modifier et adapter au fil de l'eau le comportement du robot, après la phase d'amorçage. Une modalité d'adaptation est constituée d'une part d'un moyen d'action accessible au tuteur (Interface Graphique, Joystick, Voix, etc ..) et d'autre part d'un *opérateur* capable de modifier le comportement. Dans le cadre du modèle **MPL**, un opérateur doit pouvoir agir sur la population de cellules  $\mathcal{C}_B$  représentant de manière distribuée le comportement (fig. 17.2).

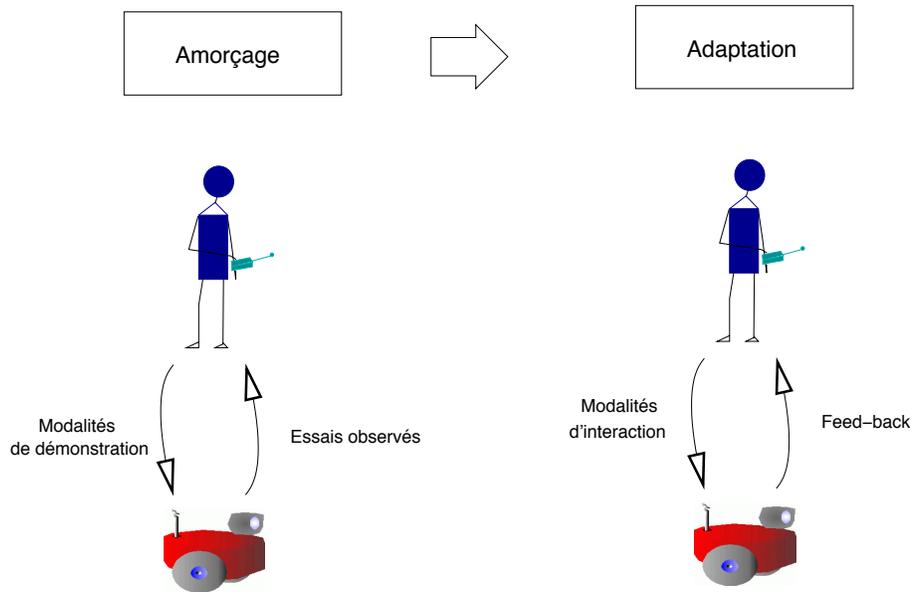


FIG. 17.1: L'apprentissage interactif permet l'adaptation du comportement et vient après une phase d'**amorçage** réalisée par démonstrations.

Les opérateurs envisageables agissent sur un sous-ensemble de cellules et jouent sur les champs  $x_i$ ,  $y_i$  et  $w_i$  des cellules ou créent (inv. suppriment) des cellules. Ils se répartissent dans l'une des catégories suivantes : ATTRACT, SAMPLE et CHANGE.

- ATTRACT : Les opérateurs de la catégorie ATTRACT consistent à modifier les cellules en les attirant vers une valeur donnée. Ils permettent par exemple de déplacer les effecteurs vers une valeur souhaitée, de déplacer les micro-percepts dans un plan perceptuel, etc ..
- SAMPLE : Les opérateurs de la catégorie SAMPLE créent ou suppriment des cellules par rééchantillonnage des données sensorielles courantes ou récentes selon un principe similaire à celui de l'algorithme *generate-pop*. Il peut s'agir par exemple, de créer des cellules supplémentaires correspondant a une zone perceptuelle désignée par le tuteur.
- CHANGE : Les opérateur de la catégorie CHANGE sont destiné à la modification de cellules existantes.

Le sous-ensemble de cellules visées par un opérateur peut correspondre à toutes sortes de critères; il peut, par exemple, s'agir des cellules actives, des cellules récemment actives, des cellules correspondant à une zone visuelle pointée par l'utilisateur via une interface graphique

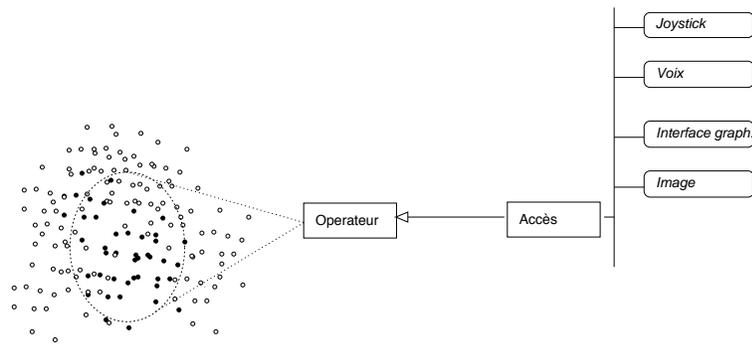


FIG. 17.2: Une modalité d'adaptation associe un opérateur d'adaptation et un moyen d'accès.

(ie, les cellules activées par cette zone de l'image).

La possibilité de formuler des opérateurs élémentaires tient à la nature de la représentation d'un comportement : chaque champ d'une cellule étant explicite, celle-ci constitue l'équivalent d'une micro-règle que l'on sait/peut modifier. Cependant, le tuteur n'a pas directement accès aux éléments individuels mais, par l'intermédiaire d'un processus éducatif, à une population d'éléments.

### 17.2.3 Etablissement d'un processus éducatif interactif

L'apprentissage interactif part du principe que le tuteur peut, par touches successives, amender le comportement initial du robot. Celui-ci, peu à peu modifié, répondra de manière différente aux situations rencontrées dans l'environnement, les changements pouvant être soit graduels, soit radicaux mais permettant qu'à son tour le tuteur construise sa propre représentation intuitive de l'effet que provoque sur le robot ses actions "pédagogiques".

Il est clair, dans un but applicatif, que l'établissement d'un tel processus éducatif, et de nature quasiment dialectique, entre le couple robot/environnement et le tuteur est nécessaire pour que celui-ci prennent confiance en les moyens d'apprentissage et puisse les utiliser au mieux. Cependant, pour qu'un tel processus puisse s'établir, il faut une rétroaction (feed back)<sup>30</sup> du robot observable par le tuteur. Cette rétroaction, peut être simplement constituée par la nouvelle réaction du robot, mais peut aussi rendre accessible au tuteur des informations plus "internes", comme la visualisation de ce que le robot perçoit ou des indicateurs de l'effet des

<sup>30</sup>cf. Travaux de l'école de Palo Alto sur l'interaction humaine et la communication [Watzlawick *et al.*, 1967].

interactions sur la population de cellules (entropie , entropie relative sachant les modifications, tendances, etc.).

## 17.3 Etude expérimentale

Une première étude de ce type d'apprentissage interactif à été réalisée par Clément Raievsky, sous ma supervision, dans le cadre de son stage de DEA. On en trouvera le détail dans le rapport [Raievsky, 2002]. L'objet de cette étude était de montrer la faisabilité de l'apprentissage interactif à partir du modèle **MPL** et de mettre en oeuvre les premiers opérateurs.

### 17.3.1 Les modalités d'adaptation mises en oeuvre

Deux modalités d'adaptation ont été mises en oeuvre. Celles-ci ont été conçues pour remédier aux deux problèmes les plus fréquemment rencontrés : le fait que les mouvements du robot doivent être parfois ajustés et le fait qu'il "hésite" dans certaines circonstances perceptuelles. Cette "hésitation" correspond à des situations où trop peu de cellules sont actives (parce le robot est devant un mur essentiellement blanc par exemple, en conséquence de l'application de la contrainte  $C(x_i)$  cf. section 11.4), situation qui entraînent une longue période de perception active (cf section 12.3). Les deux modalités sont les suivantes :

1. Une modalité d'ajustement reposant sur un opérateur de type ATTRACT conçue pour affiner les mouvements du robot en s'aidant du joystick.
2. Une modalité d'ajout d'informations reposant sur un opérateur de type SAMPLE conçue pour préciser les connaissances du robot dans des situations d'hésitations.

### 17.3.2 Modalité d'ajustement

#### Moyen d'accès

Le moyen d'accès utilisé pour cette première modalité est simplement le joystick qui permet, pendant que le robot réalise le comportement, de piloter directement le robot en en lui envoyant un ordre  $y_{tuteur}$ .

## Opérateur ATTRACT1

L'opérateur ATTRACT1 porte uniquement sur les cellules actives au moment où le tuteur effectue sa requête. Pour toutes ces cellules, il consiste à rapprocher la valeur  $y_i$  de la valeur demandée par le tuteur (eq. 17.1). Le coefficient  $\beta$  correspond au taux de modification de l'opérateur (dans les expériences  $\beta = 0.5$ ).

$$\forall cell_i \in \mathcal{S}_B(X), y_i \leftarrow (1 - \beta)y_i + \beta y_{tuteur} \quad (17.1)$$

### 17.3.3 Modalité d'ajout d'information

#### Moyen d'accès

Le moyen d'accès utilisé pour cette deuxième modalité est plus complexe. Le but recherché est de désigner une zone dans le champ visuel du robot, zone qu'il devra considérer plus particulièrement et à laquelle sera associée une action. Le tuteur réalise cette sélection en utilisant l'interface graphique (intégrée dans l'application MPL-Teach décrite dans la section 14.4.3) : lorsqu'il observe une nette hésitation du robot, il peut interrompre la phase de réalisation, désigner une zone  $Z$  rectangulaire dans l'image que voit le robot, et lui associer une action  $y_{tuteur}$  grâce au joystick. L'opérateur SAMPLE1 est alors appliqué et la phase de réalisation reprend.

#### Opérateur SAMPLE1

L'opérateur SAMPLE1 utilisé pour cette modalité consiste à créer de nouvelles cellules correspondant à la zone  $Z$  et à l'action  $y_{tuteur}$ . L'algorithme utilisé est directement dérivé de l'algorithme *generate\_pop* de la variante **MPL-s** : il crée  $m$  nouvelles cellules en échantillonnant la zone  $Z$  (alg. 3).

- 1: **pour** chacune des  $m$  cellules  $cell_i$
- 2: Choisir  $p$  positions aléatoirement dans la zone  $Z$  de l'image.
- 3: Relever les  $p$  micro-percepts correspondants et former  $x_i$ .
- 4: Créer la cellule correspondante avec  $x_i$  et  $y_{tuteur}$ .
- 5: **finpour**

**Algorithme 3:** opérateur SAMPLE1

## 17.4 Expérimentations et résultats

L'étude expérimentale à été menée sur la plateforme déjà utilisée pour l'apprentissage par démonstrations. Elle a été d'abord appliquée à des comportements jouets (décrits [Raievsky, 2002]) puis à des comportements plus complexes, comme le SLALOM.

### 17.4.1 Reprise de l'expérience SLALOM

L'intérêt de l'apprentissage interactif à partir de ces deux modalités a été clairement démontré dans le cas de l'apprentissage du comportement SLALOM déjà décrit dans la section 15.3. Cette nouvelle expérimentation à été conduite comme suit :

**Amorçage** La tâche SLALOM à été réapprise *ex nihilo* en enregistrant 5 nouvelles démonstrations volontairement peu soignées. Ceci a permis d'obtenir rapidement un premier comportement synthétique peu efficace (déviations et hésitations).

**Adaptation** Le comportement a ensuite été adapté en ligne en appliquant tour à tour les deux opérateurs disponibles. La table 17.2 donne le compte-rendu complet de l'adaptation et les étapes successives de modification du comportement. Celles-ci ont permis d'obtenir rapidement un comportement largement amélioré par rapport à celui de départ ainsi qu'à celui de la section 15.3. Les points où sont intervenus les modifications sont rappelé dans la figure 17.3.

Les résultats obtenus à la suite de cette expérience sont les suivants : le taux de réussite du comportement passe à 80% sur 20 essais (à comparer au 53% obtenus par simple démonstration). La durée de réalisation du comportement est largement réduite grâce à la suppression des situations d'hésitation (diminution par deux de la durée par rapport à l'apprentissage par démonstration).

Il est intéressant de remarquer que les deux modalités sont complémentaires et nécessaires pour mener à bien l'adaptation du comportement. Elles servent en quelques sorte d'outils pour façonner le comportement selon les besoins.

	Succès sur 30 essais
Avant interactions	53%
Après interactions	80%

TAB. 17.1: Tache Slalom : comparaison des résultat avec ou sans interactions.

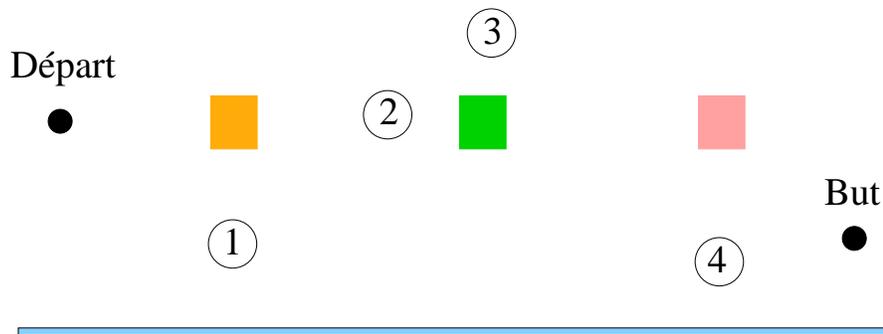


FIG. 17.3: Points d'application des operateurs interactif pour le comportement SLALOM

## 17.5 Conclusion

L'étude expérimentale de ces deux premiers opérateurs d'adaptation permet de montrer l'intérêt de l'apprentissage interactif selon les principes décrit en 17.2. On retient de cette étude que des opérateurs élémentaires peuvent permettre au tuteur d'agir sur la représentation distribuée du comportement et qu'il est possible, à partir de ces moyens, de façonner un comportement pour converger vers un comportement stable. Les opérateurs choisis permettent d'intervenir rapidement sur les quelques points non résolus par la phase d'amorçage; de ce fait le pourcentage de succès est sensiblement amélioré. Compte-tenu de son caractère simple et intuitif, le système est accessible à un utilisateur non-expert et ouvre la voie à une transmission des comportements des robots par les utilisateurs finaux. Cette étude reste encore à étendre tant sur le plan des opérateurs envisagés et leur caractérisation que sur l'ergonomie des modalités et les diverses formes de rétroactions.

J'indiquerai dans le chapitre 18 comment je pense l'intégrer dans un schéma permettant d'obtenir des comportements plus abstraits.

Etape	Réalisations successives	Opérateur	Commentaire
E1	En <b>1</b> le robot a tendance à aller tout droit.	<b>ATTRACT1</b>	altération de l'ordre moteur existant pour que le robot tourne à gauche.
E2	Vérification que la modification a été prise en compte. Le robot va toujours tout droit en <b>1</b> .	<b>ATTRACT1</b>	Idem ci-dessus.
E3	Plus de problème en <b>1</b> . Hésitation prolongée (mécanisme de perception active) en <b>2</b> et <b>3</b> . Il échoue en <b>4</b> après avoir hésité un long moment : il manque visiblement d'information à cet endroit.	<b>SAMPLE1</b>	Reéchantillonnage en <b>4</b> sur toute l'image et action de rotation vers le but
E4	Deux restitutions de suite dans lesquelles il tourne trop tôt à gauche en <b>1</b> : il touche le plot orange.	<b>SAMPLE1</b>	Reprise du comportement avant l'étape de modification précédente. On effectue un rééchantillonnage en <b>4</b> mais cette fois en sélectionnant une zone de l'image courante.
E5	Il tourne encore trop tôt à gauche en <b>1</b> et touche le plot orange mais légèrement. Une deuxième restitution à la suite est parfaite.		Pas de modification.
E6	Il se perd en <b>3</b> après une longue hésitation traduisant un manque d'information.	<b>SAMPLE1</b>	Reéchantillonnage en <b>2</b> et <b>3</b> avec sélection de zone.
E7	Deux restitution réussie de suite où il hésite en <b>4</b> mais arrive tout de même au but.	<b>SAMPLE1</b>	Rééchantillonnage en <b>4</b> avec sélection de zone.
E8	Série de 10 restitutions d'affilées où l'on obtient 8 succès complets.		Plus de modifications.

TAB. 17.2: Compte-rendu de l'adaptation du comportement SLALOM



# Chapitre 18

## Perspectives et conclusion

Ce dernier chapitre clôt ce mémoire de doctorat. Il présente d'abord les principales extensions envisageables pour **MPL**, puis expose la voie de recherche qui me semble la plus intéressante à ce point. Une conclusion générale termine le chapitre.

### 18.1 Extensions de MPL

Les principales extensions que j'envisage pour **MPL** concernent la structuration du champ perceptuel, la structuration de l'action et la généralisation de l'apprentissage interactif.

#### 18.1.1 Extension du champ perceptuel

Le robot élémentaire présenté jusqu'ici ne fait usage que d'un seul type de capteur. Une extension naturelle consiste à utiliser simultanément plusieurs capteurs (sonars, vision, laser, etc.) et à fusionner leur données dans le champ perceptuel au travers de différents plans, ainsi que le permet déjà le modèle. La thèse d'Olivier Garcia, qui consiste à concevoir des robots d'accompagnement de fantassins sur les théâtres d'opérations militaires, a pour point de départ l'étude de cette extension. Cette thèse débutera en octobre 2002, elle est co-dirigée par Alexis Drogoul et la société Thalès (Thèse CIFRE), et je participe à son encadrement. Y sera étudié en particulier, la combinaison de sonars et d'une caméra omnidirectionnelle pour projeter l'environnement sur des plans perceptuels couvrant 360 degrés et prendre en compte des informations de proximité d'obstacles.

### 18.1.2 L'articulation temporelle de l'action

Dans la section 16.3, j'ai indiqué que les comportements appris sont uniquement réactifs. Pour étendre la gamme des comportements apprenables, il est nécessaire de déterminer une structure temporelle dans les démonstrations et de prendre des décisions en fonction des actions et perceptions passées. Plusieurs voies sont envisageables pour y parvenir, il peut s'agir de relier des cellules entre elles par un lien de conséquence, de rendre les cellules récurrentes ou d'intégrer une rémanence de la perception dans les plans perceptuels. Toutes ces extensions feront l'objet de stage de DEA en 2003 et devrait permettre d'aborder des comportements plus complexes à l'avenir.

### 18.1.3 L'apprentissage interactif

Les expérimentations du chapitre 17 portant sur l'apprentissage interactif constituent clairement une pré-étude. Elles doivent maintenant être étendues en envisageant d'autres modalités et d'autres opérateurs. En particulier, il sera intéressant d'étendre les capacités de *proposition* du robot pour que celui-ci propose de son propre chef des variantes qui seront commentées par le tuteur.

## 18.2 Abstraction par interaction et réorganisation

Comme suite logique de ces différents travaux, la principale perspective que j'envisage porte sur *l'obtention de comportements abstraits* à partir de l'apprentissage interactif. Comme j'en ai déjà fait la remarque dans la section 15.5, il est nécessaire que les comportements appris soient plus abstraits pour être applicables dans un plus grand nombre de situations. Il est souhaitable en effet qu'un comportement, comme celui obtenu dans l'expérience FUSION, soit le plus possible indépendant du contexte d'apprentissage initial ; entre deux comportements dont l'un est décrit par "s'approcher d'une boîte orange lorsque celle-ci est adossée à un mur blanc" et l'autre par "s'approcher d'une boîte orange" le tuteur pourra préférer obtenir le second dans la plupart des cas. Cependant, cette opération d'abstraction, aisée sur une représentation textuelle, devient difficile dans le cas de l'apprentissage en environnement réel. Il n'est en effet pas possible de s'appuyer sur des propriétés statistiques issues des démonstrations (qui auraient toutes en commun la boîte orange) car leur nombre doit rester faible. Il n'est pas non plus possible d'utiliser un modèle de l'environnement comportant implicitement le concept de

boîte colorée (sous forme géométrique par exemple).

L'approche que je propose consiste à *associer, dans un processus incrémental, l'interaction à la réorganisation des connaissances initialement apprises*. Le sens que je donne au terme réorganisation est proche de celui pointé par Seymour Papert <sup>31</sup> lorsqu'il indique que " *Certaines des étapes les plus cruciales du développement mental sont fondées non pas seulement sur l'acquisition de nouvelles compétences, mais sur l'acquisition de nouveaux processus administratifs d'utilisation de ce que nous savons déjà.* ". De manière semblable, je pense que l'apprentissage d'un robot peut-être structuré en stades successifs permettant de réorganiser les connaissances concrètes pour les exploiter dans des situations nouvelles. L'interaction entre tuteur et robot intervient pour cadencer, orienter et diriger la réorganisation car le tuteur est le seul acteur qui puisse attribuer de la valeur à telle action du robot ou tel phénomène de l'environnement. Elle peut s'accompagner de mécanismes régulateurs (compression, réduction de l'entropie) intervenant sur l'économie interne du système.

L'idée d'un apprentissage incrémental supposant une interaction n'est pas nouvelle. Mais ici la particularité de l'approche consiste en ce que l'apprentissage est dans les termes d'une architecture distribuée, architecture où interaction et réorganisation s'appliquent non pas à quelques macro-structures mais à une population de cellules faisant intervenir un grand nombre de micro-structures.

Les processus de réorganisation envisageables sont très nombreux et restent à déterminer. Il peut s'agir de regrouper des cellules, d'agir sur un gain ou une saillance, de généraliser au sein de chaque cellule (en la considérant comme une micro-règle en logique propositionnelle), de distinguer des états au cours d'une séquence d'actions, de supprimer des cellules, d'établir entre elles des liens inhibiteurs ou de créer de nouvelles cellules chargées d'observer les autres. L'utilisation de tessels abstraits est tout à fait envisageable dans le modèle, une cellule pouvant prendre la forme suivante (cf. fig. 18.1) : *Si teinte orangée( $x_{pos1}, y_{pos1}$ )  $\wedge$   $\neg$ (teinte orangée( $x_{pos2}, y_{pos2}$ )) alors action  $y_i$  avec gain  $w_i$* . Ces processus de réorganisation correspondent aux processus de reformulation et d'abstraction décrits par Jean-Daniel Zucker [Zucker, 2001], à la nuance près qu'ils doivent être placés ici dans un cadre distribué.

---

<sup>31</sup> cité par M. Minsky dans [Minsky, 1988].

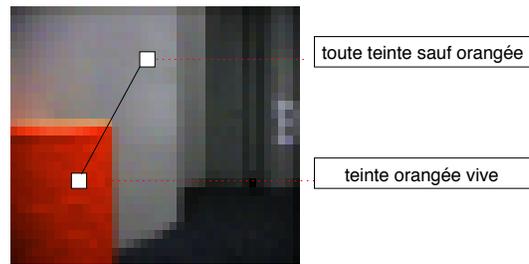


FIG. 18.1: Exemples de tessels abstraits utilisables dans une cellule

Les modalités d'interactions possibles sont dans la continuité de celles évoquées dans le chapitre 17. Elles peuvent s'inscrire dans un scénario "montrer-utiliser" dans lequel le tuteur montre les objets auquel il faut prêter attention, puis effectue l'apprentissage d'un comportement impliquant ces objets. Il peut s'agir encore pour le tuteur de marquer (par la voix, le geste) des moments importants d'une action pour que le robot en perçoive la structure.

### 18.3 Conclusion Générale

La principale contribution de ce travail a été de montrer la faisabilité de l'apprentissage de comportements d'usage en environnement réel à partir de moyens distribués élémentaires. Sur un plan interne cette faisabilité repose sur une prise en compte particulière de la perception et permet un fonctionnement qui soit implicitement de nature statistique sans passer par une représentation explicite des densités de probabilités sous-jacentes. Sur un plan externe, elle repose sur la possibilité donnée à un tuteur de montrer des exemples de comportement dans l'environnement et de garder les traces de ces interactions sous une forme réexploitable.

Au terme de ce travail, il est envisageable d'intégrer le système d'apprentissage dans une application réelle. Je pense en particulier à une application de chaise roulante intelligente à destination des handicapés, afin de faciliter leurs déplacements dans les locaux hospitaliers ou à leur domicile. Dans son principe une chaise roulante équipée d'une caméra est en effet très proche du robot élémentaire utilisé dans la phase expérimentale. L'utilisateur ou son aide médical pourrait apprendre un jeu de comportements activables en situation correspondant à des manoeuvres difficiles ou fatigantes : "entrer dans un ascenseur", "passer une porte", etc... très semblables à DOCKING et SLALOM. Plusieurs projets de chaises roulantes intelligentes existent actuellement, mais à ma connaissance tous utilisent des sonars pour reconnaître l'en-

vironnement (et principalement pour éviter des obstacles) et sont de ce fait très limitées dans leurs possibilités [Yanco, 1998],[Law *et al.*, 2002].

Comme j'en faisais la remarque dès l'introduction, la recherche robotique se trouve à la fois impliquée dans la construction de nouveaux artefacts et la tentative d'une meilleure compréhension du vivant. Si, pour des raisons méthodologiques, il faut parfois nettement séparer ces deux voies, on peut aussi espérer qu'elles s'éclairent finalement l'une l'autre. Je me suis tenu dans ce document principalement dans le rôle du concepteur d'artefact pour préserver la clarté de l'exposé mais je suis convaincu que l'étude de solutions de nature distribuée au problème de l'action adaptée des robots constitue une façon de contribuer à la compréhension de celles des être vivants.

Au-delà de l'application robotique, ce qui retient tout mon intérêt, c'est l'étude des couplages possibles entre les macro-phénomènes que constituent les comportements observables des systèmes situés et leur structure interne supposée massivement distribuée. Cette étude est aussi et nécessairement celle des moyens donnés aux humains de les façonner, les contrôler et les comprendre. J'espère dans ce document, avoir contribué à ce que cette voie de recherche fructifie dans les années à venir.



# Bibliographie

- [Agre and Horswill, 1997] P. E. Agre and I. Horswill. Lifeworld analysis. *Journal of Artificial Intelligence Research*, 6 :111–145, 1997.
- [Agre, 1988] P. E. Agre. The dynamic structure of everyday life. Technical Report AITR-1085, MIT, 1988.
- [Alami *et al.*, 1998] R. Alami, F. Ingrand, and S. Qutub. A scheme for coordinating multi-robots planning activities and plans execution. In *European Conference on Artificial Intelligence*, pages 617–621, 1998.
- [Andry *et al.*, 2001] P. Andry, P. Gaussier, S. Moga, J.P. Banquet, and J. Nadel. Learning and communication in imitation : An autonomous robot perspective. *IEEE Transaction on Systems, Man and Cybernetics, Part A : Systems and Humans*, 31(5), 2001.
- [Arkin., 1989] R.C. Arkin. Motor schema-based mobile robot navigation. *The International Journal of Robotics Research*, 8(4), August 1989.
- [Arkin, 1998] R.C. Arkin. *Behavior-based Robotics*. MIT Press, 1998.
- [Asada *et al.*, 1996] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23 :279–303, 1996.
- [Atkeson and Schall, 1997] G. Atkeson and S. Schall. Robot learning from demonstration. In J.D.H Fisher, editor, *Proceedings of the 1997 international Conference on Machine Learning (ICML'97)*, 1997.
- [Bakker and Kuniyoshi, 1996] P. Bakker and Y. Kuniyoshi. Robot see, robot do : An overview of robot imitation. In *Aisb 96 Workshop on learning in Robots and Animals*, 1996.
- [Ballard and Brown, 1993] D. Ballard and C. Brown. *Active Perception*, chapter Principles of animate vision. Lawrence Erlbaum Associates, 1993.

- [Beetz and McDermott, 1994] M. Beetz and D.V. McDermott. Improving robot plans during their execution. In *Artificial Intelligence Planning Systems*, pages 7–12, 1994.
- [Belaid and Belaid, 1992] A. Belaid and Y. Belaid. *Reconnaissance des formes, Méthodes et applications*. InterEditions, 1992.
- [Bentivegna and Atkeson, 2000] D.C Bentivegna and C.G. Atkeson. Using primitives in learning from observations. In *First IEEE-RAS International Conference on Humanoid Robotics (Humanoid-2000)*, 2000.
- [Berthoz, 1997] A. Berthoz. *Le sens du mouvement*. Editions Odile Jacob, 1997.
- [Billard and Dautenhahn, 1997] A. Billard and K. Dautenhahn. Grounding communication in situated, social robots. In *Proceedings 1997 of Towards Intelligent Mobile Robots TIMR*, 1997.
- [Billard and G.Hayes, 1999] A. Billard and G.Hayes. Drama, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behavior*, 7(1), Jan. 1999.
- [Billard *et al.*, 1998] A. Billard, K. Dautenhahn, and G. Hayes. Experiments on human-robot communication with robota, an imitative learning and communication doll robot. Technical Report CPM-98-38, Centre for Policy Modelling, Manchester Metropolitan University, 1998.
- [Billard, 1999] A. Billard. Imitation : a means to enhance learning of a synthetic proto-language in an autonomous robot. In C. Nehaniv and K. Dautenhahn, editors, *Imitation in Animals and Artifacts*. MIT Press, 1999.
- [Bishop, 1995] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford : Clarendon Press, 1995.
- [Booker *et al.*, 1989] L. Booker, D.E. Goldberg, and J.H Holland. Classifier systems and genetic algorithms. *Artificial Intelligence*, 40(1-3), 1989.
- [Bownds, 2001] M. Deric Bownds. *La Biologie de l'Esprit*. Masson Sciences. Dunod, 2001.
- [Brainard and Freeman, 1997] D. H. Brainard and W. T. Freeman. Bayesian color constancy. *Journal of the Optical Society of America*, 14, 1997.
- [Braitenberg, 1984] V. Braitenberg. *Vehicles : Experiments in Synthetic Psychology*. MIT Press, 1984.
- [Breazeal and Scassellati, 1999] C. Breazeal and B. Scassellati. A context-dependent attention system for a social robot. In *Sixteenth International Joint Conference on Artificial Intelligence (IJCAI99)*, 1999.

- 
- [Bredeche *et al.*, 2002] N. Bredeche, L. Saitta, and J.-D. Zucker. "combining wrapper and filter approaches for learning concepts from images provided by a mobile robot". In *ICML Workshop on Machine Learning in Computer Vision (MLCV-2002)*, 2002.
- [Brooks and Mataric, 1993] R.A Brooks and M. Mataric. *Robot Learning*, chapter Real Robots, Real Learning Problem. Kluwer Academic Press, 1993.
- [Brooks, 1986] R.A Brooks. A robots layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, April 1986.
- [Brooks, 1991a] R.A. Brooks. Intelligence without reason. Technical Report 1293, MIT AI Lab, 1991.
- [Brooks, 1991b] R.A. Brooks. New approaches to robotics. *Science*, (253), September 1991.
- [Bruce *et al.*, 2000] J. Bruce, T. Balch, and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. In *IROS-2000*, 2000.
- [Brunelli and Poggio, 1993] R. Brunelli and T. Poggio. Face recognition : Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10) :1042–1052, October 1993.
- [Burgard *et al.*, 2000] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D.Schulz, W. Steiner, , and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 2000.
- [Calvin, 1990] W.H. Calvin. *The Cerebral Symphony*. Bantam, New York, 1990.
- [Carson *et al.*, 1997] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Region-based image querying. In *Proceedings CVPR '97 Workshop on Content-Based Access of Image and Video Libraries*, 1997.
- [Changeux and Ricoeur, 2000] J.P. Changeux and Paul Ricoeur. *Ce qui nous fait penser*. Odile Jacob, 2000.
- [Chapelle *et al.*, 1999] O. Chapelle, P. Haffner, and V. N. Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5), September 1999.
- [Chapelle *et al.*, 2001] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 2001.
- [Connel and Mahadevan, 1993] J. Connel and S. Mahadevan. *Robot Learning*. Kluwer Academic Publishers, 1993.

- [Coradeschi and Saffioti, 2001] S. Coradeschi and A. Saffioti. Perceptual anchoring of symbols for action. In *Seventeenth International Joint Conference on Artificial Intelligence, IJCAI-2001*, 2001.
- [Cornuéjols and Miclet, 2002] A. Cornuéjols and L. Miclet. *Apprentissage artificiel - Concepts et algorithmes*. Eyrolles, Paris, 2002.
- [Crevier, 1993] D. Crevier. *A la recherche de l'intelligence artificielle*. Flammarion, 1993.
- [Cypher, 1993] A. Cypher. *Watch what I do - Programming by demonstration*. MIT Press, 1993.
- [Dautenhahn, 1995] K. Dautenhahn. Getting to know each other – artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16, 1995.
- [Dautenhahn, 1998] K. Dautenhahn. Grounding agent sociality : The social world is its own best model. In Trapp, editor, *Proceedings of the 14th European Meeting on Cybernetics and Systems*. symposium «From Agent Theory to Agent Implementation, 1998.
- [Dean and Boddy, 1988] T.L. Dean and M. Boddy. An analysis of time-dependent planning. In *Seventh National Conference on Artificial Intelligence AAAI-92*, 1988.
- [Dean et al., 1993] T. Dean, L.P. Kaelbling, J. Kirman, and A. Nicholson. Planning with deadlines in stochastic domains. In Richard Fikes and Wendy Lehnert, editors, *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 574–579, 1993.
- [Delson and West, 1993] N. Delson and H. West. Robot programming by human demonstration : subtask compliance controller identification. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, 1993.
- [Demiris and Hayes, 1996] J. Demiris and G.M. Hayes. Imitative learning mechanisms in robots and humans. In V. Klingspor, editor, *Proceedings . of the 5th European Workshop on Learning Robot*, 1996.
- [Demiris and Mataric, 1998] J. Demiris and M. Mataric. Perceptuo-motor primitives in imitation. In *Autonomous Agents'98 Workshop on Agents In Interaction - Acquiring Competence Through Imitation*, 1998.
- [Dennett, 1996] D. Dennett. *Kinds of Minds*. Brockman Inc., 1996.
- [Dietterich, 2000] T.G. Dietterich. Ensemble methods in machine learning. In J. Kittker and F. Roli, editors, *First International Workshop on Multi Classifier Systems*. NewYork : Springer Verlag, 2000.

- 
- [Dorigo and Colombetti, 1994] M. Dorigo and M. Colombetti. The role of the trainer in reinforcement learning . In *Proceeding of the Workshop on Robot Learning 1994 International Conference on Machine Learning*. MIT Press, 1994.
- [Dorigo and Colombetti, 1998] M. Dorigo and M. Colombetti. *Robot Shaping, an experiment in Behavior engineering*. MIT Press, 1998.
- [Drogoul and Collinot, 1998] A. Drogoul and A. Collinot. Applying an agent-oriented methodology to the design of artificial organizations : a case study in robotic soccer. *Autonomous Agents and Multi-Agent Systems*, 1998.
- [Drogoul and Ferber, 1992] A. Drogoul and J. Ferber. From tom thumb to the dockers : Some experiments with foraging robots. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, 1992.
- [Drogoul and Picault, 1999] A. Drogoul and S. Picault. Microbes : vers des collectivités de robots socialement situés. In *Actes des 7emes Journées Francophones Intelligence Artificielle Distribuée et Systemes Multi-Agents (JFIADSMA '99)*. Hermès, 1999.
- [Drogoul and Zucker, 2002] A. Drogoul and J.D. Zucker. Les premiers pas des robots sociaux. *La Recherche*, (350), février 2002.
- [Drogoul, 1993] A. Drogoul. *De la Simulation Multi-Agents à la Résolution Collective de pProblèmes*. PhD thesis, Université Paris VI, 1993.
- [Duda and Hart, 1973] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.
- [Dudek and Jenkin, 1999] G. Dudek and M. Jenkin. *Computational Principles of Mobile Robotics*. Cambridge University Press, 1999.
- [Edelman and Duvdevani-Bar, 1997] S. Edelman and S. Duvdevani-Bar. A model of visual recognition and categorization. *Proceedings of Royal Society*, 1997.
- [Edelman, 1992] G.E. Edelman. *Biologie de de la conscience*. Odile Jacob, 1992.
- [Edelman, 1997] S. Edelman. Computational theories of object recognition. *Trends in Cognitive Sciences*, 1997.
- [Elfes, 1989] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, june 1989.
- [Faugeras, 1993] O. Faugeras. *Three-Dimensional Computer Vision, a Geometric Viewpoint*. MIT Press, 1993.

- [Ferber, 1995] J. Ferber. *Les Systèmes Multi-Agents, vers une intelligence collective*. Inter-Editions, 1995.
- [Floreano and Mondada, 1994] D. Floreano and F. Mondada. Automatic creation of an autonomous agent : Genetic evolution of a neural-network driven robot. In *Proceedings of the Conference on Simulation of Adaptive Behavior*, 1994.
- [Foley *et al.*, 1995] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics : Principles and Practice*. Addison-Wesley, 2nd edition, 1995.
- [Fontan and Mataric, 1998] M. Fontan and M. Mataric. Territorial multi-robot task division. *IEEE Transactions of Robotics and Automation*, 14(5), 1998.
- [Friedrich and Dillman, 1995] H. Friedrich and R. Dillman. Robot programming based on single demonstration and use intention. In *Third European Workshop on Learning Robots - ECML'95*, 1995.
- [Friedrich and Kaiser, 1995] H. Friedrich and M. Kaiser. What can robots learn from humans? In *IFAC Workshop on Human-oriented Design of Advanced Robotic System*, 1995.
- [Gaussier *et al.*, 1997] 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*, 1(7), 1997.
- [Georgopoulos *et al.*, 1982] A.P Georgopoulos, J.F Kalaska, R.Caminiti, and J.T. Masszy. On the relations between the direction of two-dimensional arm movement and cells discharge in the primate motor cortex. *Journal of Neuroscience*, 2, 1982.
- [Gibson, 1986] J.J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1986.
- [Goldstone, 1998] R.L. Goldstone. Perceptual learning. *Annual Review of Psychology*, 49, 1998.
- [Graziano and Gross, 1994] M. Graziano and C. Gross. Coding of visual space by premotor neurons. *Science*, 266, 1994.
- [Harnad, 1990] S. Harnad. The symbol grounding problem. *Physica D*, 42 :335–346., 1990.
- [Hayes and Demiris, 1994] G. Hayes and J. Demiris. A robot controller using learning by imitation. In *Proceedings of the 2nd International Symposium on Intelligent Robotic System*, 1994.

- 
- [Haykin, 1999] S. Haykin. *Neural Networks, A Comprehensive Foundation*. Prentice Hall, 1999.
- [Hebb, 1949] D.O. Hebb. *The Organization of Behaviour*. John Wiley and Sons Inc., New York, 1949.
- [Holland, 1975] J. H. Holland. *Adaptation in natural artificial systems*. University of Michigan Press, 1975.
- [Horswill, 1994] Ian Horswill. Specialization of perceptual processes. Technical Report AITR-1511, MIT, 1994.
- [Hovland, 1996] G. Hovland. Skill acquisition from human demonstration using a hidden markov model. In *Proceedings IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2706–2711, 1996.
- [Hubel, 1995] D. Hubel. *Eye, Brain and Vision*. Number 22 in Scientific American Library. W.H. Freeman, New York, 1995.
- [Hugues and Drogoul, 2001a] L. Hugues and A. Drogoul. Grounding of robots behaviors. In *AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, 2001.
- [Hugues and Drogoul, 2001b] L. Hugues and A. Drogoul. Robot behavior learning by vision-based demonstrations. In *4th European workshop on advanced mobile robots (EUROBOT 01)*, 2001.
- [Hugues and Drogoul, 2001c] L. Hugues and A. Drogoul. Robot behavior learning by vision-based demonstrations. In *Colloque ALCAA, Agents Logiciels, Coopération, Apprentissage et Activité Humaine*, septembre 2001.
- [Hugues and Drogoul, 2001d] L. Hugues and A. Drogoul. Shaping of robot behaviors by demonstrations. In *First International Workshop on Epigenetic Robotics : Modeling Cognitive Development in Robotic Systems*, 2001.
- [Hugues and Drogoul, 2002a] L. Hugues and A. Drogoul. Pixel-based behavior learning. In *15th European Conference on Artificial Intelligence, ECAI'02*, 2002.
- [Hugues and Drogoul, 2002b] L. Hugues and A. Drogoul. Robotique autonome : Apprendre à partir de peu d'exemples. *Revue Cognitive*, 2002.
- [Hugues and Drogoul, 2002c] L. Hugues and A. Drogoul. Synthesis of robot's behavior from

- few examples. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'02*, 2002.
- [Hugues, 1999] L. Hugues. Rapport de stage de DEA. Technical report, LIP6 / Université Pierre et Marie Curie, 1999.
- [Hugues, 2000a] L. Hugues. Collective grounded representations for robots. In *5th International Symposium on Distributed Autonomous Robotics*, 2000.
- [Hugues, 2000b] L. Hugues. Grounded representations for a robots team. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2000*, 2000.
- [Isard and Blake, 1996] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, volume 1, pages 343–356, 1996.
- [Jakobi *et al.*, 1995] N. Jakobi, P. Husbands, and I. Harvey. Noise and the reality gap : The use of simulation on evolutionary robotics. *Lecture Notes in Computer Science*, 929, 1995.
- [Kaelbling *et al.*, 1995] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. Technical report, Brown University, 1995.
- [Kaelbling *et al.*, 1996] L. P. Kaelbling, M. L. Littman, and A. P. Moore. Reinforcement learning : A survey. *Journal of Artificial Intelligence Research*, 4 :237–285, 1996.
- [Kaiser *et al.*, 1997] M. Kaiser, V. Klingspor, and H. Friedrich. Human-agent interaction and machine learning. In *European Conference on Machine Learning*, pages 345–352, 1997.
- [Kitano *et al.*, 1997a] E. Kitano, H. Asada, Kuniyoshi M., Y. Noda, Osawa I., and H. Matsu-  
bara. Robocup : A challenge problem for ai. *AI Magazine*, Spring 1997 :73–85, 1997.
- [Kitano *et al.*, 1997b] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup :  
The robot world cup initiative. In W. Lewis Johnson and B. Hayes-Roth, editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 340–347, New York, 1997. ACM Press.
- [Klingspor *et al.*, 1996] V. Klingspor, K. J. Morik, and A. D. Rieger. Learning concepts from sensor data of a mobile robot. *Machine Learning*, 23 :305–332, 1996.
- [Kohonen, 1997] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, New York, 1997.
- [Koza, 1994] J. R. Koza. Genetic programming : On the programming of computers by means of natural selection. *Statistics and Computing*, 4(2), 1994.

- 
- [Kuniyoshi *et al.*, 1994] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching : Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6), december 1994.
- [Kurz, 1996] A. Kurz. Constructing maps for robots navigation based on ultrasonic range data. *Transactions on Systems, Man and Cybernetics*, 26(2), 1996.
- [Langton, 1989] C. G. Langton. Artificial life. In Christopher G. Langton, editor, *Artificial Life*, pages 1–47. Addison-Wesley, Redwood City, CA, 1989.
- [Law *et al.*, 2002] C.K Law, M. Leung, Yangsheng Xu, and S.K. Tso. A cap as interface for wheelchair control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'02*, 2002.
- [Lebeltel *et al.*, 2000] O. Lebeltel, P. Bessiere, J. Diard, and E. Mazer. Bayesian robots programming. *Les cahiers du laboratoire Leibniz*, 2000.
- [Linsker, 1988] R. Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21(3) :105–118, 1988.
- [Livingstone, 1989] M. Livingstone. *Les mécanisme de la vision*, chapter Art, Illusion et Système Visuel, pages 132–143. Bibliothèque Pour La Science. Belin, 1989.
- [Loeb, 1918] J. Loeb. *Forced Movements, Tropisms, and Animal Conduct*. Lippincott Company, Philadelphia, 1918.
- [Lorenz, 1978] K. Lorenz. *Les fondements de l'éthologie*. Flammarion, 1978.
- [Maes, 1990] P. Maes. Situated agents can have goals. *Journal of Robotics and Autonomous Systems*, 6 :49–70, 1990.
- [Maes, 1994] P. Maes. Modeling adaptive autonomous agents. *Artificial Life*, (1&2)(9), 1994.
- [Mahadevan and Connell, 1992] S. Mahadevan and J. Connell. Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence*, 55 :311–365, 1992.
- [Marr, 1982] D. Marr. *Vision : a Computational Investigation into Human Representation and Processing of Visual Information*. W.H. Freeman and Company , NY 26-61, 1982.
- [Mataric, 1992] M.J. Mataric. Integration of representation of into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3), 1992.

- [McCarthy and Hayes, 1969] J. McCarthy and P. Hayes. Some philosophical problems from the stand point of artificial intelligence. *Machine Intelligence*, 4, 1969.
- [Meltzoff and Moore, 1992] A.N. Meltzoff and M.K. Moore. Early imitation within a functional framework : the importance of person identity, movement and developpement. *infant behavior developpement*, 15, 1992.
- [Merleau-Ponty, 1945] M. Merleau-Ponty. *Phénoménologie de la perception*. Gallimard, 1945.
- [Merleau-Ponty, 1964] M. Merleau-Ponty. *L'Oeil et l'Esprit*. Gallimard, 1964.
- [Meyer and Guillot, 1989] J.A. Meyer and A. Guillot. Simulation of adaptative behavior in animats : Review and prospect. *From Animals to Animats*, 1989.
- [Meyer and Guillot, 1994] J.-A. Meyer and A. Guillot. Four years of animat research. In Cliff D., Husbands P., Meyer J.-A., and Wilson S., editors, *From animals to animats 3*, pages 2–11, Cambridge, MA, 1994. MIT Press.
- [Meyer, 1995] J.A. Meyer. The animat approach to cognitive science. In H. Roitblat and J.A. Meyer, editors, *Comparative Approaches to Cognitive Science*. The MIT Press, 1995.
- [Minsky, 1988] M. Minsky. *La société de l'esprit*. InterEditions, 1988.
- [Mitchell, 1980] T.M. Mitchell. The need for biases in learning generalisation. Technical report, Rutgers Technical Report, 1980.
- [Mitchell, 1997] T.M. Mitchell. *Machine Learning*. Mc Graw Hill, 1997.
- [Moore *et al.*, 1995] A. Moore, C. Atkeson, and S. Schaal. Memory-based learning for control. Technical Report CMU-RI-TR-95-18, CMU Robotics Institute, 1995.
- [Moravec, 1982] H. Moravec. The cmu rover. In *Second national artificial intelligence conference, AAAI-82*, 1982.
- [Moravec, Summer 1988] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, Summer 1988.
- [Nehmzow and Smithers, 1991] U. Nehmzow and T. Smithers. Mapbuilding using self-organizing networks. In J.A. Meyer and S. Wilson, editors, *From Animals to Animats*. MIT Press, 1991.
- [Nehmzow, 1999] U. Nehmzow. Vision processing for robot learning. *Journal of Industrial Robot*, 26(2), 1999.

- 
- [Nehmzow, 2001] U. Nehmzow. Quantitative analysis of robot-environment interaction- on the difference between simulation and the real thing. In *4th European workshop on advanced mobile robots (EUROBOT 01)*, 2001.
- [Nicolescu and Mataric, 2001] M. Nicolescu and M.J. Mataric. Learning and interacting in human-robot domains. *IEEE Transactions on Systems, Man, Cybernetics*, 31(5), 2001.
- [Nilson, 1973] N. Nilson. A hierachical robot planning and execution system. Technical report, Standford Research Institute - Artificial Intelligence Center, Menlo Park, 1973.
- [Nilsson, 1984] N. Nilsson. Shakey the robot. Technical Report 323, SRI, Menlo Park, Calif., 1984.
- [Norman, 1988] D. A. Norman. *The Psychology of Everyday Things*. Basic Books, New York, 1988.
- [Nourbakhsh *et al.*, Summer 1995] I. Nourbakhsh, R.Powers, and S. Birchfield. Dervish an office-navigating robot. *AI Magazine*, 16(2), Summer 1995.
- [Osuna *et al.*, 1997] Edgar Osuna, Robert Freund, and Federico Girosi. Support vector machines : Training and applications. Technical Report AIM-1602, 1997.
- [Pal and Pal, 1993] N. Pal and S. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9) :1277–1294, 1993.
- [Pfeifer, 1996] R. Pfeifer. Symbols, patterns and behavior : towards a new understanding of intelligence. In *Japanese Conference on Artificial Intelligence*, 1996.
- [Pfeifer, 1998] R. Pfeifer. Embodied system life. In *Proceedings of the International Symposium on System Life*, July 1998.
- [Piaget, 1977] J. Piaget. *La naissance de l'intelligence chez l'enfant*. Delachaux et Niestlé, 1977.
- [Poggio and Edelman, 1990] T. Poggio and S. Edelman. A network that learns to recognize 3d objects. *Nature*, 1990.
- [Poggio and Koch, 1989] T. Poggio and C. Koch. *Les mécanisme de la vision*, chapter Des synapses qui détectent le mouvement, pages 85–96. Bibliothèque Pour La Science. Belin, 1989.
- [Pomerlau, 1993] D.A Pomerlau. Knowledge-based training of artificial neural networks for autonomous robot driving. In J.Connell and S. Mahadevan, editors, *Robot Learning*, 1993.

- [Pomerlau, 1997] D.A Pomerlau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 15(88), 1997.
- [Raievsky, 2002] C. Raievsky. Opérateurs interactifs pour l'éducation d'un robot mobile. Technical report, Université Paris VI, 2002.
- [Rechenberg, 1973] Ingo Rechenberg. *Evolutionstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann–Holzboog, Stuttgart, 1973.
- [Riesenhuber and Poggio, 2000] M. Riesenhuber and T. Poggio. Models of object recognition. *Nature*, 3, november 2000.
- [Rizolatti *et al.*, 1996] G. Rizolatti, L. Fadiga, V. Gallese, and L Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3, 1996.
- [Russ, 1995] J.C. Russ. *The Image Processing Handbook 2nd Edition*. Boca Caton,FL :CRC Press, 1995.
- [Russel and Norvig, 1995] S. Russel and P. Norvig. *Artificial Intelligence - a modern approach*. Prentice Hall International, 1995.
- [Ruwet, 1975] J.C. Ruwet. *Ethologie : biologie du comportement*. Pierre Mardaga, éditeur, 1975.
- [Sallantin, 1994] J. Sallantin. *Ordre biologique, ordre technologique*, chapter Ce que peut apprendre une machine. collection milieux. CHAMP VALLON, 1994.
- [Sallantin, 1997] J. Sallantin. *Les agents rationnels*. Hermès, 1997.
- [Schwefel, 1975] Hans-Paul Schwefel. *Evolutionstrategie und numerische Optimierung*. Dr.-Ing. Thesis, Technical University of Berlin, Department of Process Engineering, 1975.
- [Sempe *et al.*, 2002] F. Sempe, A. Munoz, and A. Drogoul. Autonomous robots sharing a charging station with no communication : a case study. In *6th International Symposium on Distributed Autonomous Robotic Systems ,DARS 2002*, 2002.
- [Simmons and Koenig, 1995] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of IJCAI-95*, 1995.
- [Simonin and Ferber, 2001] O. Simonin and J. Ferber. Modélisation des satisfactions personnelle et interactive d'agents situés coopératifs. In *9eme journées Francophones d'Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, pages 215–226, 2001.

- 
- [Smithers, 1994] T. Smithers. On why better robots make it harder. In *From Animals To Animats 3 : SAB'94*, 1994.
- [Sonka *et al.*, 1993] M. Sonka, V. Haclav, and R. Boyle. *Image Processing, Analysis and machine Vision*. Chapman ad Hall, 1993.
- [Steels, 1999] Luc Steels. How language bootstraps cognition. In *Proceedings der 4. Fachtagung der Gesellschaft fur Kognitionswissenschaft*, 1999.
- [Stone, 1993] J.V. Stone. Computer vision : What is the object? In *Artificial Intelligence and Simulation of Behavior*, Birmingham, 1993.
- [Stricker and Orengo, 1995] M. Stricker and M. Orengo. Similarity of color images. In *SPIE'95*, 1995.
- [Suchman, 1987] S. Suchman. *Plans and Situated Actions : The Problem of Human-Machine Communication*. Cambridge University Press", 1987.
- [Sutton, 1984] R. S. Sutton. Temporal credit assignment in reinforcement learning. Technical Report 84-02, 1984.
- [Swain and Ballard, 1991] M.J. Swain and D.H Ballard. Color indexing. *International Journal of Computer Vision*, 7(1), 1991.
- [Thrun *et al.*, 1997] S. Thrun, A. Bucken, W. Burgard, D. Fox, T. Frohlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schmidt. Map learning and high-speed navigation in rhino. In R.R Murphy D. Kortenkamp, R.P. Bonnasso, editor, *AI-based Mobile Robots : Case studies of successfull robot systems*. MIT Press, 1997.
- [Thrun *et al.*, 1999] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Daellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva : A second-generation museum tour-guide robot. In *Proceedings IEEE International Conference on Robotics and Automation*, 1999.
- [Thrun *et al.*, 2001] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence Journal*, 2001.
- [Thrun, 1997] S. Thrun. To know or not to know : On the utility of models in mobile robotics. *Ai Magazine*, 18(1), 1997.
- [Tinbergen *et al.*, 1942] N. Tinbergen, B.JA Meeuserand L.K. Boerema, and W.W. Rossieau. Die bals des samtfalters, euminis semele. *Tierpsychol.*, 5, 1942.

- [Tootel *et al.*, 1982] R.B. Tootel, M.S. Silverman, E. Switkes, and R.L. Valois. Desoxyglucose analysis of retinotopic organization in primate cortex. *Science*, 218 :902–904, 1982.
- [Turner, 1998] R.M. Turner. Context-mediated behavior for intelligent agents. *International Journal of Huma-Computer Studies*, 48(3), 1998.
- [Uexkull, 1965] J.V. Uexkull. *Mondes animaux et monde humain*. Médiations Denoel, 1965.
- [Vapnik, 1999] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1999.
- [Warren, 1995] W.H. Warren. *Global perspectives on the ecology of human-machine systems*, chapter Constructing an econiche. Lawrence Erlbaum, 1995.
- [Watt and Policarpo, 1998] A. Watt and F. Policarpo. *The Computer Image*. Addison-Wesley, 1998.
- [Watzlawick *et al.*, 1967] P. Watzlawick, J.H. Beavin, and D.D. Jackson. *Une logique de la communication*. Edition du seuil, 1967.
- [Wilson, 1991] S. W. Wilson. The animat path to AI. In Jean-Arcady Meyer and Stewart W. Wilson, editors, *From Animals To Animats 1 : SAB'91*, pages 15–21, 1991.
- [Wolpert and Macready, 1997] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1) :67–82, April 1997.
- [Worrall and Baker, 1993] A. Worrall and S. Baker. Advances in model based traffic vision. In *British Machine Vision Conference*, 1993.
- [Yamauchi and Langley, 1997] B. Yamauchi and P. Langley. Place recognition in dynamics environnements. *Journal of Robotic Systems*, 14(2), 1997.
- [Yan *et al.*, 2001] J. Yan, R. Wood, S. Avandhanula, M. Sitti, and R.S. Fearing. Toward flapping wing control for a micromechanical flying insect. 2001.
- [Yanco, 1998] H.A. Yanco. "wheesley, a robotic wheelchair system : indoor navigation and user interface.". *Lecture Notes in Artificial Intelligence : Assistive Technology and Artificial Intelligence*, 1998.
- [Zell *et al.*, 1992] A. Zell, N. Mache, R. Huebner, M. Schmalsl, T. Sommer, and T. Korb. (snns) : Stuttgart neural network simulator. Technical report, University of Stuttgart, 1992.
- [Ziemke, 1999a] T. Ziemke. *Does Representation Need Reality*, chapter Rethinking Grounding. Plenum Press, 1999.

---

[Ziemke, 1999b] T. Ziemke. *Recurrent neural networks : design and applications*, chapter Remembering how to behave : Recurrent neural network for adaptative robot behavior. Boca Raton : CRC press, 1999.

[Zucker, 2001] J.D. Zucker. Changement de representation, abstractions et apprentissage. Technical report, Université Pierre et Marie Curie, Décembre 2001.