HAL
open science

La représentation contingente : vers une réconciliation des approches fonctionnelles et structurelles de la robotique autonome

Eric Dedieu

To cite this version:

Eric Dedieu. La représentation contingente : vers une réconciliation des approches fonctionnelles et structurelles de la robotique autonome. Interface homme-machine [cs.HC]. Institut National Polytechnique de Grenoble - INPG, 1995. Français. NNT : . tel-00005043

HAL Id: tel-00005043

https://theses.hal.science/tel-00005043

Submitted on 24 Feb 2004

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

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

THÈSE

présentée à

L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE (INPG)

**Laboratoire d'Informatique Fondamentale
et d'Intelligence Artificielle (LIFIA)**

pour obtenir le titre de

docteur en informatique

par

Éric DEDIEU

La représentation contingente

vers une réconciliation des approches fonctionnelles et structurelles de la robotique autonome.

Soutenue le 12 septembre 1995 devant le jury composé de :

MM.	**Augustin**	**LUX**	Président
	Emmanuel	**MAZER**	Directeur
	John	**STEWART**	Rapporteurs
	Paul	**BOURGINE**	
	Stevan	**HARNAD**	Examinateurs
	Philippe	**JORRAND**	

Remerciements

Emmanuel Mazer, Pierre Bessière et Olivier Lebeltel ont formé le “noyau dur” des discussions animées qui ont présidé tous les aspects de cette thèse. Leur motivation et leur capacité d'écoute ont été le moteur de mes réflexions, et je les remercie d'autant plus de leur patience que je ne leur ai pas toujours facilité la tâche… Un grand merci à Manu de m'avoir supporté tant de temps comme thésard et de m'avoir laissé suivre mon chemin hors des sentiers battus. Et que Rémis Balaniuk, Juan-Manuel Ahuactzin-Larios, Bernard Amy, Arnaud Giacometti, et tous les autres qui ont fait que le climat de travail au Lifia a toujours été agréable et enrichissant, trouvent également ici les remerciements qu'il méritent.

Je ne saurais trop exprimer ma gratitude à Jim Crowley, Christian Laugier, Patrick Reignier, Emmanuel Mazer, et à tous ceux qui se sont décarcassés pour permettre mon financement. C'est grâce à eux que j'ai pu mener à bien cette thèse, et j'ai eu beaucoup de chance d'avoir été aidé ainsi.

N'oublions pas Pascal Di-Giacomo et Philippe Bobet, dont les conseils techniques et services divers ont été bien utiles et bien appréciés.

Enfin, un grand merci à John Stewart et Paul Bourgine d'avoir accepté d'être rapporteurs et d'avoir apprécié mes idées, et avec eux à Stevan Harnad, Philippe Jorrand et Augustin Lux pour m'avoir fait l'honneur de constituer mon jury et de juger de mon travail.

Cette thèse a été financée par une bourse CNRS (BDI) et un complément de bourse INRIA.

Résumé

La robotique autonome s'attache à éliminer (ou du moins à minimiser) l'intervention du concepteur humain dans le fonctionnement d'un robot en environnement "complexe". Or, la programmation d'un robot repose toujours sur l'utilisation de modèles dont le domaine de validité est assez restreint. Quand la situation concrète rencontrée par le robot sort de ce domaine de validité, on tombe sur le problème de l'**imprévu**, objet de cette thèse.

Les préoccupations qui suivent sont illustrées dans ce document par une série d'expériences robotiques très simples mais d'une nature inhabituelle. Par son aspect fondamental, cette thèse se situe plus dans les sciences cognitives que dans le domaine habituel de la robotique.

Nous affirmons en premier lieu :

1) que l'autonomie d'un robot ne peut être obtenue sans une gestion systématique de l'imprévu

   Les humains savent gérer l'imprévu, pas les robots. Et en pratique, les méthodes habituelles de programmation des robots reposent sur la disponibilité d'un concepteur capable de gérer l'imprévu à la place des robots. Le rôle de ce concepteur est souvent de modifier l'environnement pour éliminer la cause de l'imprévu et rétablir la validité de la modélisation utilisée. Légitime et efficace en robotique industrielle, cette pratique ne convient plus à la robotique autonome qui vise à développer des robots dans des environnements **naturels**, c'est-à-dire indépendants de la manière dont le robot est conçu. L'imprévu doit à terme être géré par la machine. L'utilisation de techniques neuronales ou de comportements réflexes ne change rien à ce constat.

2) que l'approche dominante en intelligence artificielle (IA) ne permet pas d'aborder cette question de l'imprévu

   Pour rendre un robot autonome on utilise habituellement des modèles très généraux, censés de par leur abstraction représenter une grande variété d'environnements, par exemple des modèles géometriques. On oublie que de tels modèles ne sont généraux que par la liberté qu'ils laissent à l'humain: tout problème concret a de nombreux aspects non géometriques et donc non gérés par le modèle, et c'est à l'humain d'user de ses propres facultés et de son "sens commun" pour contraindre l'environnement de façon à ce que seuls les aspects géometriques posent un problème à résoudre par le robot. L'utilisation de ces modèles sans une telle intervention humaine est illusoire, et ils ne sont donc pas adéquats pour aborder le problème fondamental de l'autonomie.

Pour aborder ces difficultés nous proposons trois pistes :

1) Sur le plan théorique, nous défendons la nécessité d'une reconnaissance explicite par un robot de sa propre ignorance, et donc d'une gestion systématique de l'incertitude. Nous utilisons pour cela une théorie de la logique et du raisonnement probabilistes (Jaynes 1995). Cette théorie a pour nous le meme rôle fondateur que la logique formelle peut avoir pour l'IA traditionnelle.

2) Sur le plan méthodologique, nous affirmons que la distinction classique entre les cycles de conception (des structures permanentes) et d'adaptation (des paramètres variables) doit être complétée par un cycle d'incrementalité (systématisation d'une évolution structurelle en réponse à certains imprévus). Nous mettons ainsi l'accent sur le développement opportuniste du robot plus que sur la recherche de méthodes "exactes" et universelles permettant de programmer un robot pour réaliser certaines tâches. Notre démarche se distingue des démarches classiques de l'incrementalité, d'une part par le rôle créatif de l'imprévu, considéré comme le moteur du développement, et d'autre part par la volonté d'en systématiser certains aspects.

3) Sur le plan conceptuel, nous proposons de revoir la notion classique de représentation. Une représentation est habituellement définie par sa "fonction", c'est-à-dire la façon prédéterminée dont elle peut être utilisée. Nous proposons au contraire une notion de "représentation contingente", qui separe capacité de représentation et interprétation. La capacité de représentation est intrinsèque à une structure, tandis que l'interprétation devient dépendante du contexte. On a ainsi une définition structurelle de la représentation. Il s'agit là d'un changement radical dans la façon de formuler la question de la représentation, qui permet notamment de mettre l'accent sur l'origine et la génèse des représentations plutôt que sur leurs performances fonctionnelles. Les critiques de l'approche classique avaient conduit certains chercheurs (Varela 1988) à rejeter la notion même de représentation pour aborder l'autonomie, mais avec ce rejet on abandonne un guide de conception incontournable. La représentation contingente, échappant à ces critiques, offre la perspective d'aborder le problème de la conception au sein d'approches jusqu'ici non exploitées en IA (par exemple la notion de clotûre opérationnelle).

Table des matières

Remerciements **3**

Résumé **5**

Table des matières **7**

Chapitre 1. DES SCIENCES COGNITIVES À LA ROBOTIQUE AUTONOME **11**

A. Introduction **11**

B. À la croisée de différents domaines **13**

C. Plan du document **19**

D. Notes sur la forme du document **22**

Chapitre 2. PROBLÉMATIQUE **23**

A. Les deux cycles de développement traditionnels **24**

A.1. Définitions : robot, structure, paramètres 24
A.2. La séparation entre conception et adaptation 25

B. Le cycle de conception **26**

B.1. L'analyse descendante 27
B.2. L'analyse comportementale 29
B.3. La notion de modularité 29

C. Critique du cycle de conception en robotique autonome **31**

C.1. Le problème des conditions de validité 31
C.2. Le problème de la modularité 36

D. Le cycle d'adaptation : présentation et critique **38**

D.1. L'apprentissage 39
D.2. A-t-on résolu le problème des conditions de validité ? 40

E. Notre réponse aux problèmes soulevés jusqu'ici **42**

E.1. La notion d'incertitude 43
E.2. Le cycle de l'incrémentalité 44

F. La représentation prédéfinie **51**

F.1. La notion de représentation prédéfinie 52
F.2. Une “cause commune” à tous les problèmes ? 55

G. Conclusion : notre problématique **58**

H. Compléments **59**

H.1. Le danger des métaphores biologiques 59
H.2. Le danger des simulations 61
H.3. Note historique et terminologique 64

Chapitre 3. INCERTITUDES ET PROBABILITÉS **67**

A. Logique et probabilités **67**

A.1. Le théorème de Cox 67
A.2. De la logique déductive à la logique probabiliste 70
A.3. Conventions de notation et formules utiles 71

B. Le principe du maximum d'entropie **73**

B.1. Le principe d'indifférence 74
B.2. Le résultat principal du principe du maximum d'entropie 76
B.3. Un cas particulier utile : les distributions gaussiennes 77

C. Décision et action **77**

D. Application à la robotique **78**

D.1. Terminologie 78
D.2. Systématicité des calculs 80

E. Autres théories du raisonnement incertain **80**

Chapitre 4. APPRENTISSAGE **83**

A. L'expérience de la bassine lumineuse **83**

A.1. Dispositif expérimental 84
A.2. Modélisation probabiliste 86
A.3. Inférences réalisables sur le modèle probabiliste 88

B. Un système d'inférence probabiliste **90**

B.1. Définition et but 90
B.2. La connaissance C 90
B.3. Les mécanismes d'inférence 93
B.4. Implémentation informatique 95

C. Résultats : incertitudes et conditions de validité **96**

C.1. Les caprices d'une cellule photo-électrique 96
C.2. Le choix d'un modèle : hypothèses ou connaissances ? 98
C.4. Le cas des environnements inconnus 101

D. Inférence par réseaux bayésiens **102**

E. L'expérience du Khépéra téléguidé **104**

Chapitre 5. INCRÉMENTALITÉ **107**

A. Dépendances empiriques probabilistes (DEP) **107**

A.1. Définition 108
A.2. Variables primitives 108

B. L'incrémentalité : illustrations expérimentales **110**

B.1. L'expérience de la bassine perturbée 110
B.2. L'expérience de la mire 121
B.3. L'expérience du Khépéra photophile 128

C. Particularités du cycle incrémental **129**

C.1. Les DEP doivent systématiquement être interprétées 130
C.2. Comportements professeurs 130
C.3. La gestion de l'imprévu 131
C.4. Incertitude et interprétation 132
C.5. La gestion des variables artificielles 133

Chapitre 6. LA REPRÉSENTATION CONTINGENTE **137**

A. une notion de représentation non prédéfinie **137**

B. comparaison à d'autres approches de la représentation **140**

B.1. Comparaison avec l'approche de Lloyd (1988) 140
B.2. Comparaison avec la “deictic representation” de Agre (1988) 142

C. des robots “pour quoi faire” ? **143**

C.1. Les buts du robot : une notion émergente 143
C.2. Viabilité et autonomie : une perspective 145

Chapitre 7. CONCLUSION **147**

A. Synthèse 147
B. Vers une réconciliation… ?.....148
C. Perspectives.....149

Annexe 1 153
Annexe 2 155
Bibliographie 157

CHAPITRE 1

DES SCIENCES COGNITIVES À LA ROBOTIQUE AUTONOME

Il y a trop de notes.
L'empereur dans “Amadeus”

A. Introduction

Cette thèse a pour ambition ultime d'automatiser en robotique la gestion de ce que nous appelons *l'imprévu.* Le problème est le suivant. Un robot a été programmé pour se comporter d'une certaine façon dans son environnement, or un observateur constate qu'il ne se comporte pas comme prévu, par exemple qu'il n'effectue pas la tâche prévue, ou qu'il le fait avec des effets parasites imprévus, voire même qu'il résoud des problèmes imprévus. *Que faire alors devant ces constatations ?*

Illustrons ce problème par deux exemples survenus dans notre laboratoire, auxquels nous nous réfèrerons par la suite comme “le syndrome des pas de porte” et “l'histoire du KitBorg”.

Le “syndrome des pas de portes” est typique du genre de misère que l'on rencontre dans les expérimentations robotiques. Un robot mobile, dont la tâche était de se rendre de la salle robotique au hall d'entrée, suivait une trajectoire planifiée à partir d'une carte du rez-de-chaussée. En passant la porte séparant la salle du hall, une roue a patiné sur la bosse que faisait le pas de porte en cuivre, et la trajectoire en ligne droite s'est révélée mener droit dans un mur. À ce point l'opérateur a tout stoppé pour éviter la collision. L'imprévu était ici la présence de ce pas de porte et son influence sur le comportement du robot.

“L'histoire du KitBorg” est un peu plus inhabituelle. Le KitBorg est un petit robot d'environ 15 cm de haut et de diamètre. Il est commandé en vitesse par deux roues motrices, et dispose de quatre cellules photoélectriques situées devant, derrière, à droite et à gauche. Ces cellules ne sont calibrées que très grossièrement. Le KitBorg était programmé selon le principe suivant. On compare en permanence les valeurs des cellules avant, gauche et droite, et la commande dépend de la cellule la plus excitée : si c'est la cellule avant le robot avance tout droit, si c'est la cellule gauche il dévie sur la gauche, si c'est la cellule droite il dévie sur la droite. Le robot a donc tendance à se diriger vers la lumière. Typiquement, il peut être dirigé par le faisceau d'une torche électrique.

Moins typiquement, le KitBorg peut éviter les obstacles, négocier son passage dans une porte entrouverte, et faire signe qu'il a atteint son but ! Cela a été remarqué lors de tests dans les locaux de la société qui développait ce robot. Celui-ci suivait le faisceau d'une torche. On s'aperçut avec surprise que lorsque dans ces conditions un obstacle lui barrait la route, le KitBorg l'évitait naturellement. Et quand on éteignait la torche, le robot déviait vers une porte de bureau entrouverte, passait cette porte, et arrêtait son périple sous l'ampoule du bureau où il se mettait à “danser”… Malgré la maladresse des mouvements, cette interprétation de la trajectoire complexe du robot en termes de navigation dans une scène s'imposait à tous les témoins, bien

que sans rapport avec la fonction explicitement programmée. Celle-ci était censée faire se diriger le robot vers la lumière… mais ce faisant, tout obstacle assez grand projette une ombre qui incite à l'éviter, et l'éclairage d'un bureau diffuse un cône de lumière par la porte qui a tendance à y attirer le robot, et la configuration des lieux permettait à ces phénomènes de bien se manifester. La “danse” sous l'ampoule, elle, était dûe aux mouvements devenus aléatoires dans le “marais” lumineux à cet endroit ; en d'autres occasions ces mouvements auraient été jugés “incohérents”, mais en l'occurence la pensée s'est naturellement imposée à chacun que le robot “est content parce qu'il a atteint son but”…

On peut dire que cette anecdote est à l'origine de la thèse présentée ici. Elle suscite de nombreuses interrogations. Tous les témoins ont vu le robot contourner les obstacles ; pourtant, connaissant la façon dont il était programmé, on pourrait dire “qu'en fait” il ne contournait pas les obstacles mais les ombres… et encore, “en fait” il ne contournait pas réellement les ombres, son programme ne faisant aucune référence aux ombres. Tous ces “en fait” traduisent une ambiguïté profonde entre ce que faisait le robot a-posteriori pour les observateurs humains, et ce qu'il était censé faire a-priori pour son programmeur. Il y a là un rapport intéressant à étudier voire à exploiter, et c'est le thème de cette thèse.

Le syndrome des pas de porte et l'histoire du KitBorg illustrent deux formes très différentes de ce que peut être l'imprévu en robotique. La question qui se pose ensuite, c'est de prendre des mesures, ayant constaté un imprévu, pour changer et améliorer la programmation du robot a-posteriori. C'est un problème de taille, dont une quelconque résolution automatique effective reste chimérique actuellement bien qu'il s'agisse de l'ambition “de principe” de nos travaux. Dans le cadre de cette thèse, nous voulons plus modestement présenter une *méthodologie* permettant de prendre en compte ce problème de l'imprévu. Cette méthodologie s'appuye sur deux notions : *l'incrémentalité* et *la représentation.*

La méthodologie incrémentale (“bottom-up” en anglais) s'oppose à la méthodologie descendante (“top-down”). Cette dernière est la façon classique d'aborder un problème : elle consiste à le décomposer en sous-problèmes, puis à décomposer chaque sous-problème récursivement, jusqu'à obtenir une hiérarchie de problèmes que l'on peut alors résoudre individuellement (modularité) avant de les faire participer à la résolution globale. La démarche incrémentale, au contraire, ne se fonde pas sur la donnée préalable d'un problème global donné, mais sur les capacités brutes d'un robot que l'on va *développer.* On commence par résoudre les problèmes les plus simples que l'on sache résoudre, sans se préoccuper de spécifier une décomposition hiérarchique a-priori. Ensuite, en se fondant sur les capacités élémentaires ainsi acquises, on complexifie le robot de façon plus ou moins opportuniste, afin de l'amener au comportement global désiré. Plutôt qu'une décomposition a-priori d'un problème en sous-problèmes, on assiste ainsi à la construction a-posteriori d'une solution à partir de solutions partielles.

La seconde notion clef de cette thèse est celle de représentation. Nous voyons cette notion comme le moyen de systématiser la démarche incrémentale, qui dans sa pratique actuelle revêt encore un caractère totalement ad-hoc. Cette idée est inhabituelle, la notion de représentation étant souvent au contraire rejetée par les approches incrémentales. Mais selon nous, ce n'est qu'une certaine approche de la représentation qui est rejetée. Nous en proposons dans cette thèse une approche différente, que nous appelons “la représentation contingente”, et que nous pensons plus à même de convenir à la démarche incrémentale.

Le caractère globalement assez conceptuel de cette thèse reflète l'idée que les difficultés actuelles de la robotique ne sont pas seulement techniques mais en grande partie conceptuelles : il ne suffira pas d'attendre une certaine maturité technologique pour que les nombreuses méthodes dont la justification est restée théorique se trouvent aptes à une utilisation pratique, et en ce sens la sophistication de plus en plus poussée des architectures et des méthodes utilisées apporte en définitive très peu de progrés concrets. Il va falloir, pensons-nous plutôt, *poser* les problèmes en d'autres termes, abandonner certains désidérata et en imposer d'autres.

B. À la croisée de différents domaines

Dans l'introduction ci-dessus, nous ne nous sommes pas clairement situés dans un domaine de recherche. Nous avons cité la robotique, mais ce n'est pas réellement le cadre adéquat pour aborder les questions soulevées (imprévu, incrémentalité, représentation). En effet, il s'agit de comprendre une partie du rôle habituellement dévolu au concepteur humain dans la construction de robots — pour ultimement systématiser voire automatiser ce rôle. À ce titre, ces questions ne concernent pas seulement la robotique, mais s'inscrivent également dans les domaines de “l'intelligence artificielle” et des “sciences cognitives”. Nous décrivons à présent notre conception de ces différents domaines, pour adopter en conclusion la “robotique autonome” comme cadre de réflexion.

Sciences cognitives

On trouve dans la littérature diverses définitions pour les sciences cognitives. La plupart sont des définitions en extension, c'est-à-dire une simple énumération de disciplines. Bien peu d'auteurs se risquent à une définition en compréhension, ce qui se comprend étant donné la jeunesse[1] et la multidisciplinarité de ce domaine, la complexité de son objet d'étude, la variété des points de vue et des niveaux d'analyse possible, et l'ambition formidable de son propos.[2]

> «Le fait est qu'à l'heure actuelle il n'y a pas de consensus global, parmi la communauté des chercheurs en sciences cognitives, sur un cadre théorique qui permettrait de répondre de manière précise à cette question. Dans les termes de Kuhn (1962), la situation est typiquement pré-paradigmatique : les opinions les plus diverses foisonnent. Ainsi, pour certains, l'objet des sciences cognitives est en premier lieu la rationalité humaine, avec les extensions que permettent les ordinateurs ; pour d'autres il s'agit de toutes les connaissances exprimables dans le langage ; et pour d'autres encore la cognition n'est pas réservée aux êtres humains, mais existe également chez les animaux. Une approche par énumération des disciplines n'est guère plus concluante. Si un large consensus existe pour considérer que la psychologie cognitive, la linguistique et l'intelligence artificielle en font partie, déjà la place des neurosciences peut être contestée par certains […] alors qu'elle est considérée comme fondamentale par d'autres. Et au delà de ces “quatre grands”, l'ensemble est singulièrement

---

[1] Le domaine d'étude des sciences cognitives n'est pas à proprement parler nouveau. La problématique de l'intelligence ou celle du comportement humain ou animal sont même très anciennes. L'élément neuf qui justifie de présenter les sciences cognitives comme un domaine encore jeune est qu'on dispose à présent de moyens technologiques qui permettent d'expérimenter, d'observer, de simuler, et de formaliser des questions jusque là restées beaucoup plus spéculatives.

[2] De toute façon, peu de domaines scientifiques peuvent être définis en compréhension. Même la physique, par exemple, n'apparaît comme un domaine clairement cerné que grâce à de nombreux siècles d'histoire et de pratique, qui font que l'usage du mot s'est peu à peu précisé.

flou. L'anthropologie, l'épistémologie, l'éthologie, l'informatique, la logique, la psychologie sociale, la sociologie de la connaissance, lui appartiennent-elles ou non ?» (John Stewart, *Intellectica n° 16,* 1993, p. 7-8)

Pour notre part, l'aspect qui nous intéresse dans les sciences cognitives est la recherche et l'étude de processus expliquant l'interaction des êtres vivants et artificiels avec leur environnement. C'est cela que nous appellerions la “cognition”, en remplacement du terme “intelligence” devenu trop vague et chargé de sens multiples.

Nous voyons donc les sciences cognitives non comme un programme de recherche ou un projet scientifique multidisciplinaire déjà bien défini, mais avant tout comme un *domaine d'intérêt* présentant de multiples facettes. En l'absence d'une problématique commune claire, alors que la véritable question reste encore d'identifier les problèmes pertinents, ces facettes restent liées à des disciplines scientifiques séparées, représentant autant d'angles d'attaque potentiellement intéressants. Ce qui justifie le rapprochement de ces diverses disciplines est une interaction réelle des communautés, la recherche d'une interdisciplinarité qui peut être l'embryon d'une perspective scientifique des plus ouvertes.

«C'est parce que notre édifice semble un peu bricolé que nous sommes si exigeants sur la qualité des pierres…» (Yves Gueniffey, *Intellectica n°17,* 1993, p. 241)

Intelligence artificielle

L'intelligence artificielle (IA) n'est guère définie plus clairement que les sciences cognitives. Historiquement, l'IA a longtemps et fortement été dominée par le courant de pensée dit “cognitiviste” (la manipulation de symboles comme credo, la logique et les systèmes formels comme théorie fondatrice, l'ordinateur comme métaphore du cerveau), au point que nous avons longtemps considéré le cognitivisme comme définissant l'IA. Cependant, beaucoup de travaux actuels rejettent cette conception dominante tout en se réclamant de l'IA ; on dit alors “IA classique” pour faire référence aux travaux de type cognitiviste.

Certaines définitions ont été proposées, mais qui restent loin d'être universellement acceptées :

- L'IA est caractérisée par la prééminence de l'explicabilité sur l'efficacité ; un programme d'IA non seulement produit des résultats mais explique de façon compréhensible la démarche suivie pour cela (Kodratoff [cours de DEA de F. Rechenman])
- L'IA s'intéresse à la résolution automatique des problèmes qui sont couramment résolus par les êtres vivants bien que formellement extrêmement complexes, par exemple des problèmes NP-complets (Jean-Marc Fouet, discussion personnelle).

Nous nous plaçons plutôt dans cette seconde perspective, généralisée comme *l'abord de la problématique de la cognition par la réalisation et l'étude de systèmes artificiels,* c'est-à-dire en somme la branche artificielle des sciences cognitives. L'IA ainsi définie regroupe de nombreux travaux largement entremêlées :

- La résolution de problèmes de nature symboliques ou logiques, i.e. la branche “classique” : systèmes experts, démonstration automatique, planification en monde clos ;
- L'intelligence artificielle distribuée ;
- Le connexionnisme et les réseaux de neurones formels ;
- Les systèmes hybrides (intégration de techniques connexionnistes et symboliques) ;
- La dynamique des systèmes complexes et l'auto-organisation ;
- La vie, évolution ou écologie artificielles.

Robotique

Le domaine de la robotique semble clairement défini. Il s'agit d'assembler des effecteurs, des capteurs, des dispositifs mécaniques, électroniques et informatiques, de façon à automatiser certaines tâches.

La robotique est un cadre de travail particulièrement contraignant, car certains problèmes y sont incontournables. Avant de faire marcher la moindre expérience de robotique, il faut résoudre une foule de problèmes techniques tels que calibration, réglages mécaniques ou électroniques, transmissions, positionnements, etc… Ces problèmes prennent du temps et donnent l'impression d'entraver les recherches sur les véritables centres d'intérêt. Pour cette raison, beaucoup de travaux en sont restés au stade de la simulation et n'ont pas été réellement implantés et testés sur des robots physiques. Cette “paresse” est justifiée a-posteriori en argumentant du manque de temps et de moyens, et surtout en entretenant l'illusion que les difficultés pratiques seraient minimes et purement techniques. De ce fait, on trouve souvent le terme “robot” employé comme une métaphore dans des travaux concernant exclusivement des simulations informatiques.

Pour notre part, nous pensons que les difficultés pratiques chroniques de la robotique ne doivent pas être ainsi éludées. Ce sont elles qui font la spécificité de la robotique par rapport à l'IA, et qui caractérisent le fait de travailler dans un monde réel plutôt que dans une simulation. Loin donc d'entraver les recherches sur les véritables centres d'intérêt, ce sont ces difficultés pratiques qui sont fondamentales et dont il faut comprendre la nature. Par conséquent, nous ne prendrons jamais la robotique dans un sens métaphorique, mais au sens propre de l'utilisation de robots physiques dans un environnement physique. Ce choix sera argumenté plus longuement au chapitre 2.

Même avec ces réserves, le domaine de la robotique reste très large, et on peut le voir selon plusieurs angles :

- *La nature des environnements considérés.* Le contrôle de l'environnement peut faire partie des moyens sur lesquels peut jouer le concepteur du système. En robotique industrielle, la spécification des situations rencontrables et le contrôle de l'environnement font partie des moyens autorisées. Au contraire, on peut chercher à faire fonctionner les robots dans des environnements non contrôlés, connus seulement de façon incertaine et incomplète.
- *Les tâches que le robot doit accomplir.* On peut vouloir un robot spécialisé pour une tâche bien définie, ou spécialisé pour un ensemble de tâches, ou bien assez

souple pour se voir progressivement rajouter des tâches non spécifiées lors de la conception initiale.

- *L'efficacité et les performances demandée au robot.* Elles peuvent être plus ou moins essentielles, selon qu'on développe un robot industriel (le but est la performance), qu'on explore certaines techniques non encore maîtrisées (le but est une avancée technologique dont les retombées industrielles ne seront pas immédiates), ou qu'il s'agit de recherche fondamentale (le but est alors plus scientifique que technique, et l'aspect technique peut se restreindre à montrer la faisabilité ou le potentiel d'une approche dont les retombées industrielles ne sauraient être qu'à très long terme).
- *Réactivité et contrôle.* Un robot peut avoir plusieurs modes de fonctionnement, c'est-à-dire plusieurs façons de gérer un flux de données. On peut s'intéresser au maintien à tout instant d'un mode donné (aspect “contrôle”, gestion de flux de données), ou aux transitions entre modes (aspect “réactif”, changements d'état globaux).

Robotique autonome : les environnements physiques et naturels

Une chaîne de montage industrielle, la salle robotique d'un laboratoire le jour d'une démonstration, cette même salle robotique à un moment “normal”, une salle de séjour, une route goudronnée dans un campus, la jungle amazonienne : tous ces environnements sont physiques (non simulés), mais de natures très différentes. Un robot de montage industriel n'est pas intuitivement un robot “autonome”, tandis qu'un robot capable de se déplacer dans la jungle amazonienne se verrait considérer comme hautement autonome. Où se situe donc la limite ?

Les réponses techniques à cette question sont peu satisfaisantes. L'essence des difficultés nous semble être appauvrie dès que l'on choisit de catégoriser un environnement comme plus ou moins bruité ou dynamique, ou selon le nombre de paramètres qui le décrivent. Pour notre part, nous choisissons plutôt de distinguer les environnements comme plus ou moins *naturels.* Un environnement naturel est donné au robot indépendemment de la conception propre de celui-ci, tandis qu'un environnement artificiel est partiellement conçu par rapport au robot.

Par exemple, un robot mobile destiné à fonctionner dans une salle informatique est dans un environnement artificiel si pendant le fonctionnement du robot les tables et chaises sont remises bien en place, et si les étudiants ont pour consigne de ne pas se mettre dans le champ de perception du robot et de ne pas allumer telle ou telle lumière. Le robot est par contre dans un environnement naturel si son entrée en scène n'impose aucune restriction particulière à l'état “normal” de la pièce. S'il s'agit du hall d'entrée du laboratoire — autre exemple — les personnes doivent pouvoir s'y déplacer, la présence de cartons ou de meubles ne doit pas y être bannie “pour l'occasion”, etc…

Malgré ces nombreuses possibilités, une salle informatique ou le hall d'entrée d'un laboratoire restent des environnements naturellement peu hostiles et somme toute très stables. Les passants font naturellement attention au robot même s'ils s'amusent probablement beaucoup à le tester ; il n'existe pas de “prédateurs” naturels ; la température reste dans une gamme jamais dangereuse pour le robot ; les murs, portes et lampes ne changent pas de place…

Une salle informatique ou un hall ne se définissent pas naturellement par des propriétés spécifiques, des règles de fonctionnement. Il est de inutile de préciser formellement toutes ces propriétés : *c'est un état de fait.* Partir du principe contraire rendrait l'environnement “artificiel” à notre sens. Il faut au contraire faire agir le robot “en situation” (“situated action”, Suchman 1986, Agre 1988) — et le fait même qu'il s'agisse d'une salle informatique va bien entendu faire en sorte que les conditions (informelles) généralement rencontrées dans de tels environnements soient respectées.

Cet axe naturel/artificiel est indépendant de l'axe physique/simulé pour caractériser un environnement. On peut imaginer un monde simulé et naturel : comme exemple citons le jeu vidéo “Pengo” auquel joue le programme “Pengi” de Agre & Chapman (1987). L'environnement est ce qu'il est ; si le programme fait des postulats sur de quelconques règles de fonctionnement et si ces postulats lui permettent de gagner, ce ne sera pas parce que la dynamique de l'environnement en aura été affectée autrement que par l'action du programme dans celui-ci.

> *Définition* : nous appelons “robotique autonome” la robotique en milieu physique *et* naturel. Plus que des critères purement techniques (incertitudes, dynamique, complexité de l'environnement), ce qui sépare la robotique autonome de la robotique non autonome est donc le rôle du concepteur dans l'environnement.

Cette acception de “robotique autonome” s'oppose à plusieurs autres[3] telles que :

- Un robot est autonome s'il effectue des tâches variées, dans un environnement dynamique, sans contrôle direct par le concepteur.
- Un robot est autonome s'il a des buts et sait trouver les actions à accomplir pour les satisfaires sans que ces actions lui aient été prescrites explicitement.
- Un robot est autonome s'il est capable, en fonction de son environnement, de changer son propre comportement, c'est-à-dire soit ses buts soit le moyen d'atteindre ces buts

Nous n'avons pas pour notre part défini ce qu'est un robot autonome, mais plutôt ce qu'est la robotique autonome en tant que domaine de recherche. Nous pensons en effet qu'il est encore trop tôt pour se poser la question de ce qu'est un robot autonome, ne serait que parce que nous sommes trop loin d'avoir réellement compris la question pour être sûrs de bien la formuler. Notamment, nous avons tenu à éviter toute référence à une notion de but, dont nous nous méfions beaucoup (nous en reparlerons). Nous préférons donc nous en tenir à un domaine de recherche (la robotique dans un environnement naturel) plutôt que proposer un projet bien défini (la réalisation d'un robot ayant telle ou telle propriété).

Mais alors, comment savoir si notre objectif est atteint ? Et pouvons-nous affirmer faire de la robotique autonome sans être capables de reconnaître un robot autonome si nous en voyons un ?

Tout d'abord, notre objectif à court terme est plus de *bien poser le problème* de la robotique autonome que de le résoudre. Nous souhaitons mettre en évidence

---

[3] Pour dissiper les malentendus, nous avons cherché un temps un terme de rechange pour “autonomie”, sans rien trouver de satisfaisant. Nous prions donc le lecteur qui trouverait ce terme inadéquat de le comprendre néanmoins dans notre sens et non dans un de ses sens plus habituels.

certaines questions habituellement éludées, justifier qu'elles sont dignes d'être creusées, et proposer des pistes conceptuelles, méthodologiques, théoriques et techniques pour les aborder.

Ensuite, si nous ne savons pas ce qu'*est* un robot autonome, nous avons des idées sur ce que ce *n'est pas.* Le chapitre suivant détaillera abondamment notre point de vue à ce sujet. Selon nous, il n'existe pas actuellement de robots autonomes, et l'échéance nous semble assez lointaine pour que le problème d'une définition précise ne se pose pas encore.

Enfin, la question “positive” de l'autonomie peut actuellement être abordée selon deux points de vue entre lesquels nous ne souhaitons pas trancher *a-priori* :

- Le premier définit l'autonomie d'un organisme (vivant ou artificiel) comme son aptitude à réaliser certaines tâches dans son environnement. Ces tâches peuvent être la navigation en environnement encombré, la capacité de réagir à certaines modifications imprévues de celui-ci, l'organisation automatique d'actions pour atteindre un but, la robustesse à divers aléas, etc… C'est la réalisation de ces tâches qui est présentée comme le critère d'autonomie (ou condition de viabilité). Cette approche, que nous appellerons “fonctionnelle”, est la plus répandue en robotique.
- En parallèle, il existe en sciences cognitives un courant de recherche récent sur une notion d'autonomie non orientée par la notion de tâche, mais par la notion de “clôture opérationnelle” (Varela 1989). Grossièrement, l'idée est de définir l'autonomie d'un organisme au niveau des *mécanismes* de fonctionnement de celui-ci et non à un niveau fonctionnel, et de redonner en cela à “autonomie” son sens étymologique de “qui définit ses propres lois”. Cette approche, que nous qualifierons de “structurelle”, est surtout explorée par des biologistes (Varela 1989, Stewart 1993a), et n'a encore eu aucun impact pratique en robotique, bien que souvent citée comme source d'inspiration.

Ces deux positions sont généralement présentées comme opposées, en compétition. En robotique cet antagonisme est sans doute une mauvaise chose. Pour nous, si l'approche structurelle n'a eu aucun impact pratique en robotique, c'est que l'approche fonctionnelle reste incontournable quand il s'agit de *concevoir* des robots, et non de décrire des organismes vivants existant déjà. Et, réciproquement, si l'approche fonctionnelle ne connaît aucun progrès en pratique malgré la sophistication des techniques utilisées, c'est peut-être qu'il est temps de prendre en compte certaines idées de l'approche structurelle. C'est pour cette raison que nous ne souhaitons pas choisir a-priori sur la nature de l'autonomie des robots — et cherchons même, en fin de compte, à réconcilier les aspects structurels et les aspects fonctionnels de la robotique autonome.

Ceci étant dit, par rapport aux notions d'imprévu, d'incrémentalité et de représentation, le choix de la robotique autonome comme cadre d'étude s'impose pour plusieurs raisons :

- D'une part, la robotique autonome conduit naturellement à poser le problème de l'imprévu. Ce qui pour nous caractérise les environnements physiques et naturels, c'est qu'ils ne sont par essence pas formalisables, et donc le problème de l'imprévu s'y pose *systématiquement.*

- D'autre part, au sein des sciences cognitives la robotique autonome est seule à poser vraiment le problème de la *conception* de systèmes autonomes. C'est ce qui la distingue de disciplines telles que la biologie, l'éthologie ou la linguistique, qui ont pour but de comprendre certains systèmes cognitifs existants, et non de les construire effectivement. Or, la notion d'incrémentalité est essentiellement liée à cette optique de conception, ainsi dans une certaine mesure que la notion de représentation.
- Enfin, étant donné l'aspect conceptuel de nos travaux, nous tenons à garder un lien avec la robotique. Cette référence concrète est essentielle pour guider notre réflexion et juger de sa portée. Toutes nos réflexions seront exposées et justifiées dans le cadre de la programmation des robots, même si toute perspective d'application industrielle est pour l'instant lointaine.

C. Plan du document

Dans ce document nous développons côte à côte trois types de réflexions :

- Une réflexion critique sur les approches actuelles de la robotique, conduisant à remettre en cause et reformuler certaines notions importantes telles que conception, modélisation, environnement, apprentissage, interprétation, automatisation, fonctionnalités ou buts… Cela fait l'objet du chapitre 2.
- Une réflexion théorique et technique permettant de formaliser nos préoccupations et nos propositions. Cela nous a menés à l'utilisation d'une théorie du raisonnement probabiliste comme alternative aux systèmes formels et à la logique à deux états.[4] Cette théorie est décrite au chapitre 3. La mise en oeuvre technique et algorithmique reste embryonnaire mais offre des perspectives intéressantes, elle fait l'objet du chapitre 4.
- Une réflexion méthodologique et conceptuelle sur la programmation des robots, fondée sur une reconsidération des rôles respectifs du concepteur humain et de l'expérimentation. Nous mettons l'accent sur l'importance du développement incrémental dans la conception d'un robot, sur la prise en compte de la notion d'imprévu, et nous proposons une notion originale de représentation permettant de guider cette conception. Ces sujets sont présentés aux chapitre 5 et 6 à partir d'expériences robotiques d'un type inhabituel.

Voici à présent un résumé des chapitres 2 à 6 (le chapitre 7 étant la conclusion).

Chapitre 2 : problématique

Le chapitre 2 retrace les réflexions qui nous ont amenés à focaliser notre intérêt sur les notions d'incrémentalité et de représentation. Ce faisant il replace notre problématique dans le contexte actuel de la robotique autonome tel que nous le voyons, et offre une analyse critique des approches actuellement dominantes dans le domaine.

Plus précisément, nous dégageons dans celles-ci trois problèmes :

---

[4] La proposition du raisonnement probabiliste comme une véritable théorie fondatrice pour l'IA ne sera pas le sujet principal de cette thèse, mais restera présente en arrière-plan. Pour un développement plus poussé de ce thème, voir le travail de Pierre Bessière (1995, non encore publié).

- *Le problème des conditions de validité.* Lorsqu'on automatise une tâche concrète par un certain mécanisme, le fait que ce mécanisme résolve effectivement cette tâche n'est jamais acquis. Il reste toujours, à côté des aspects qu'on a su modéliser et automatiser, des aspects qui doivent être gérés activement par le concepteur humain, faisant appel au sens commun, à l'imagination, à l'intelligence générale. C'est ce que nous appelons les “conditions de validité” du modèle. Or, chercher à éliminer l'intervention du concepteur dans un environnement naturel, c'est chercher à automatiser la gestion des conditions de validité, et cela mène à des difficultés fondamentales.
- *Le problème de la modularité.* Le principe de toute conception est de décomposer un problème en sous-problèmes pouvant ensuite être résolus séparément et indépendemment (modularité). Mais l'indépendance des sous-problèmes n'est jamais acquise en pratique ; la modularité est plus un guide de conception que quelque chose que l'on peut obtenir systématiquement. Or les approches habituelles de la programmation des robots partent du principe d'une modularité effective, et cela se révèle inadéquat lorsqu'il s'agit d'éliminer l'intervention du concepteur dans l'environnement.
- *Le problème de l'imprévu.* Nous avons déjà parlé en introduction du problème de l'imprévu : que faire lorsque le robot ne fait pas ce qu'il est censé faire ? Comment corriger, ou même exploiter, cette constatation ? Avant d'éliminer l'intervention du concepteur dans l'environnement, il faut s'interroger sur la nature de l'imprévu en robotique.

Nous analysons alors ces problèmes comme provenant de deux sources :

- *Une source méthodologique.* Le développement d'un robot est souvent séparé en deux étapes : celle de conception proprement dite (structure matérielle et logicielle du robot, capteurs et actionneurs), et celle d'adaptation (identification de paramètres en fonction de données expérimentales). Nous critiquons cette séparation comme inadéquate à la robotique autonome.
- *Une source conceptuelle.* La construction d'un robot est toujours fondée sur une notion de représentation que nous appelons “prédéfinie”. Cette notion dit que ce qui est invariant dans une représentation, c'est sa fonction — la façon dont cette représentation est utilisée. Nous prétendons que *pour la robotique autonome* ce n'est pas une notion adéquate, même si elle peut le rester pour l'ingénieur ou le psychologue.

Deux pistes nous semblent permettre d'aborder ces problèmes :

- L'incrémentalité, c'est-à-dire le développement de la *structure* même du robot en réponse à son comportement expérimental, et plus précisément comme méthode d'adaptation à l'imprévu.
- La prise en compte d'une notion d'incertitude, vue ici comme une manifestation de “l'ignorance” du robot. Les modèles utilisés pour programmer un robot ne sont jamais parfaits — y compris les modèles empiriques tels que les réseaux de neurones —, et nous défendons la nécessité d'une formalisation explicite de l'incertitude, qui permettrait à un robot de juger expérimentalement de l'adéquation des modèles utilisés et de les utiliser en conséquence.

Nous abordons également dans ce chapitre, certains sujets annexes : le danger de l'emploi inconsidéré de simulations en robotique, et le danger des métaphores biologiques auxquelles font parfois appel certains travaux.

Chapitre 3 : la logique probabiliste

Nous abordons au chapitre 3 la formalisation de la notion d'incertitude dont nous aurons ressenti la nécessité au chapitre 2. Nous avons pour cela adopté la théorie du raisonnement probabiliste “probability as logic” du physicien E.T. Jaynes (1994). Le fondement de cette théorie est le théorème de Cox, qui propose la notion mathématique de probabilité comme une formalisation de la notion intuitive de “plausibilité”. Cela permet de proposer la théorie des probabilités comme une extension de la logique booléenne, et comme une théorie de l'inférence et du raisonnement incertain.

Chapitre 4 : le cycle d'adaptation

Le chapitre 4 utilise la théorie présentée au chapitre 3 pour réaliser un système probabiliste d'apprentissage et de gestion de l'incertitude.

Ce système d'apprentissage est illustré par son application à une expérience élémentaire, “la bassine lumineuse”, qui décrit comment nous faisons caractériser un environnement lumineux à une cellule photoélectrique mobile autour d'un axe. La même approche est ensuite appliquée à une seconde expérience, “le Khépéra téléguidé”, qui décrit comment selon les mêmes principes nous faisons apprendre un comportement d'évitement d'obstacle à un robot mobile Khépéra.

Les résultats expérimentaux, assez surprenants par certains côtés, nous permettent alors d'une part de dégager l'intérêt d'une gestion explicite de l'incertitude, et d'autre part d'expliquer la distinction essentielle entre les notions d'hypothèses et de connaissances.

Chapitre 5 : le cycle de l'incrémentalité

Le chapitre 5 présente des expériences plus complexes que les précédentes, expériences “en plusieurs étapes” mettant en oeuvre notre notion d'incrémentalité.

Plus précisément, nous voyons l'incrémentalité comme le développement expérimental d'un ensemble de structures de données particulières appelées “dépendances empiriques probabilistes” (DEP). La particularité d'une DEP est qu'ayant été construite dans un contexte d'apprentissage particulier, elle est néanmoins exploitable par la suite dans d'autres contextes, et cette exploitation nécessite de la part du concepteur un acte de *réinterprétation* de la “signification” de la DEP dans le nouveau contexte.

L'objectif principal des expériences du chapitre 5 est d'illustrer ce processus d'aquisition de DEP suivi d'une exploitation par réinterpréation, notamment pour aboutir à la formation de nouvelles DEP et permettre un certain type d'incrémentalité.

La première expérience est une sophistication de la bassine lumineuse, où il s'agit de reconnaître la position et la couleur de bandes de papier posées sur le bord de la bassine. La seconde expérience consiste à bouger une caméra face à une mire afin de caractériser leur mouvement relatif, puis d'estimer par ce biais la distance de la mire par rapport à la caméra. La troisième expérience est inspirée de l'histoire du

KitBorg, et consiste à faire apprendre un comportement d'évitement d'obstacle par le biais d'un comportement initialement photophile.

Chapitre 6 : la représentation contingente

Pour synthétiser les leçons du chapitre 5 sur un mode plus conceptuel, nous proposons dans le chapitre 6 une notion de représentation inhabituelle car non-prédéfinie : la “représentation contingente”.

La réapparition de la notion de représentation, tant décriée en général par les approches incrémentales, s'explique par la motivation suivante : si l'on veut pouvoir systématiser un processus de développement incrémental, il faut à chaque étape préciser ce qui, dans la structure d'un robot, peut être amené à changer, et ce qui au contraire doit rester invariant pour ne pas perdre l'expérience passée que l'on souhaite encore pouvoir exploiter. La notion de représentation est pour nous le moyen de faire cette distinction, de désigner ce qui doit rester constant et pour quelle raison. Une représentation interne, c'est une trace de l'expérience passée, exprimée dans des termes propres au robot, qui puisse être génériquement exploitable pour la programmation future du robot.

La représentation contingente consiste à nettement séparer la notion de représentation et celle d'interprétation : une représentation contingente est systématiquement interprétable, *mais le sens qu'on lui accorde (son interprétation) est variable et contingente au contexte.* Cette notion de représentation est compatible avec les approches structurelles de l'autonomie, car l'invariant qu'elle propose est une structure et non une fonction. Cependant, le lien avec les approches fonctionnelles, si nécessaires à la conception, n'est pas coupé, et se trouve dans la notion d'interprétation.

Nous proposons alors les DEP comme un exemple élémentaire de représentation interne.

Nous développons aussi dans ce chapitre ce que devient la notion de “but” dans cette approche, car ses rôles traditionnels ne conviennent pas à une approche non strictement fonctionnelle.

D. Notes sur la forme du document

Un chapitre est divisé en *sections* (A, B, etc.). Les divisions d'une section sont nommées *paragraphes* (A.1, A.2, etc.). La notation §A.1 renvoie à un paragraphe, du même chapitre si aucun chapitre n'est spécifié.

Nous avons “adopté” les anglicismes suivants, commodes car sans équivalent simple en français : *ad-hoc* au sens “conçu spécifiquement pour résoudre un problème donné et donc non systématisable”, *alternative* au sens “solution de rechange”.

CHAPITRE 2

PROBLÉMATIQUE

Il s'était fait mal dans la rue
mais on l'a soigné autre part.
Boby Lapointe

Nous avons esquissé notre motivation générale comme : “aborder le problème de l'imprévu en robotique autonome par une démarche incrémentale”. Mais en vérité ce n'est pas cette question qui à l'origine a guidé nos travaux, disons plutôt qu'elle en est un résultat, qu'elle est née d'une réflexion générale sur les difficultés des approches actuelles de la robotique autonome. Si nous en sommes venus à nous focaliser sur la problématique particulière de l'imprévu et de l'incrémentalité, c'est que cela nous a paru un bon point de départ pour aborder certaines de ces difficultés.

Ce chapitre retrace nos réflexions, et replace donc notre problématique dans le contexte actuel de la robotique autonome tel que nous le voyons. Il offre en même temps une analyse critique des approches actuellement dominantes dans le domaine.

Voici le plan détaillé de ce long chapitre, divisé en sections A à H :

- **Section A.** Pour aborder la construction d'un robot, il est en général admis de séparer celle-ci en deux étapes : d'une part la conception de la structure physique et logicielle du robot, et d'autre part, si cette structure comprend des mécanismes adaptatifs, l'adaptation expérimentale effective du robot dans un environnement particulier. Dans la section A nous détaillerons (et admettrons dans un premier temps) cette séparation conception-adaptation.
- **Sections B et C.** Nous décrirons dans la section B la démarche classique pour la conception, et montrerons dans la section C que son efficacité repose sur l'existence permanente d'un concepteur humain capable de résoudre les problèmes informels non pris en compte dans la démarche de conception elle-même. Nous verrons que c'est de cette intervention humaine que la robotique autonome cherche à s'affranchir en partie, et que cela pose certaines difficultés fondamentales.
- **Sections D.** Nous nous demanderons alors dans la section D si un cycle d'adaptation peut suffire à résoudre les difficultés que rencontre le cycle de conception. Nous concluerons que ce n'est pas le cas.
- **Section E.** Les sections C et D auront soulevé quelques difficultés dans la démarche conception-adaptation proposée au chapitre A. La section E proposera alors deux notions grâce auxquelles nous pensons aborder ces difficultés : les notions d'incertitude et d'incrémentalité. La première est liée au cycle d'adaptation. Étant donné un modèle, nous appelons incertitude la confiance que l'on accorde à ce modèle et à ses prédictions, et nous défendons la nécessité d'une gestion explicite de cette incertitude. L'incrémentalité, quant à elle, constitue un troisième cycle dans la construction d'un robot, que nous proposons d'insérer entre conception et adaptation. Nous proposons de plus de fonder ce cycle incrémental

sur une notion de “représentation interne”. Nous montrons enfin en quoi nous considérons cela comme une réponse au problème de l'imprévu présenté au chapitre 1.

- **Section F.** La section F développe l'idée de fond pour laquelle nous pensons que la notion de représentation est une voie intéressante pour aborder l'incrémentalité. Nous montrons pour cela que toutes les difficultés de la robotique autonome décrites jusqu'ici peuvent être considérées comme les conséquences de l'adoption universelle d'une notion de représentation particulière dite “représentation en termes prédéfinis”. Par conséquent, la recherche d'une notion alternative peut être effectivement la clef que nous cherchions pour aborder ces problèmes.
- **Section G.** La section G fait une synthèse des sections précédentes pour faire, parmi tous les sujets soulevés, un choix sur ceux que nous allons plus précisément aborder dans cette thèse.
- **Section H.** Il reste trois réflexions importantes, qui ont leur place dans ce chapitre mais qui ne font pas partie de l'argumentation principale (disons qu'ils servent à l'illustrer), et que nous avons groupées dans une dernière section H. D'abord, nous critiquerons l'appel à certaines métaphores biologiques abusives en IA ou en robotique autonome. Nous exposerons ensuite les dangers de l'utilisation de simulations en robotique autonome, et la façon de les éviter. Ensuite, nous comparerons l'analyse critique que nous aurons menée dans ce chapitre à la façon dont ces critiques sont habituellement exprimées dans les travaux qui se placent dans le même courant.

A. LES DEUX CYCLES DE DÉVELOPPEMENT TRADITIONNELS

On distingue souvent, dans l'élaboration d'un robot, deux étapes que nous appellerons “conception” et “adaptation”. La conception concerne la structure fixe (matérielle et logicielle) du robot, et l'adaptation concerne ce qui peut être modifié par l'expérience, et donc adapté aux circonstances particulières où se trouve le robot. Il s'agit là de l'application à la robotique d'une méthodologie courante dans les sciences de l'ingénieur.

A.1. Définitions : robot, structure, paramètres

Un robot peut se définir comme un système physique en interaction avec son environnement, capable de caractériser cette interaction via ses capteurs, et de la modifier via ses actionneurs.

À tout moment, l'état des actionneurs est caractérisé par des grandeurs propres au robot appelées “variables motrices”, et l'état des capteurs par des grandeurs propres appelées “variables sensorielles”. L'ensemble de toutes ces variables, globalement appelées “sensori-motrices”, est ce qui dans les termes propres au robot caractérise son état instantané. Le robot ne peut donc pas distinguer des situations instantanées que distinguerait un éventuel observateur extérieur mais qui

résulteraient dans un même ensemble de valeurs pour les variables sensori-morices.[1]

Le mécanisme responsable du comportement du robot peut être tant matériel (morphologie, asservissements, électronique, nature physique des capteurs ou actionneurs…) que logiciel (algorithmes, structures de données…). Nous y distinguerons deux composantes :

- *Sa structure,* qui est l'ensemble des caractéristiques fixes qui ne peuvent être contrôlées arbitrairement par le robot lui-même,
- *Ses paramètres,* qui sont toutes les grandeurs arbitrairement contrôlables.

Notamment :

- Les caractéristiques physiques font partie de la structure. Elles peuvent être amenées à changer, mais par l'intermédiaire de l'action (accidents, vieillissement) et non par un contrôle arbitraire.
- Un mécanisme d'asservissement fait partie de la structure. La consigne de cet asservissement fait partie des paramètres si elle est arbitrairement modifiable, de la structure sinon.
- Les instructions informatiques (algorithmes) qui constituent le programme du robot font partie de la structure.
- Le choix des variables sensori-motrices et des paramètres fait partie de la structure (ce sont leurs *valeurs* qui peuvent changer).
- Supposons que le programme du robot utilise un algorithme connexionniste. Le plus souvent, l'ensemble des unités est fixe ainsi que les connexions qui les relient : ils font partie de la structure. Les poids des connexions, qui peuvent être contrôlés, constituent les paramètres.
- Plus généralement, les algorithmes utilisent souvent des variables modifiables arbitrairement au cours du temps, qui peuvent être vues comme la “mémoire” du robot. Elles sont considérées comme des paramètres.

Dans la définition ci-dessus, la structure du robot est tant logicielle que matérielle. La structure matérielle est en général peu susceptible de changer entre les applications, et on peut souvent se ramener à un point de vue purement informatique, où la conception se résume à la programmation du robot, et où les paramètres considérés sont ceux du programme. La partie analogique du robot est alors mise de côté. Ce point de vue informatique simplifie la discussion sans en restreindre la portée, et nous l'adopterons par la suite.

A.2. La séparation entre conception et adaptation

En termes de structure et paramètres, précisons à présent ce que nous entendons par conception et adaptation :

---

[1] L'état est quelque chose d'instantané : nous ne prenons donc pas en compte la notion de temps. Par exemple, pour manipuler indépendemment les valeurs d'une variable A à l'instant présent et à un instant inférieur, nous devons en faire deux variables différentes A_0 et A_1. En pratique nous éviterons ce genre de procédé, étant conscients que ce n'est pas une bonne façon de gérer l'aspect temporel.

- Le rôle du concepteur est de donner la structure du robot (cycle de conception), qui comprend entre autre la liste des paramètres et la façon dont ils se modifient avec l'expérience.
- L'identification expérimentale des paramètres à certaines valeurs particulières permet l'adaptation du robot à une situation particulière (cycle d'adaptation).

Conception et adaptation sont des problèmes de natures différentes, et il semble qu'il soit fructueux de les aborder séparément dans un premier temps :

- *Aborder le problème de l'adaptation en faisant abstraction de celui de la conception.* Prenons l'exemple des diverses techniques neuronales permettant d'apprendre des associations stimulus-réponse. Il est clair que pour appliquer concrètement ces techniques au contrôle d'un robot, il faut que les grandeurs que l'on associe soient pertinentes pour la tâche que l'on cherche à effectuer. Ce problème “de conception” est alors supposé déjà résolu indépendemment, et on ne cherche à étudier que les propriétés adaptatives intrinsèques des réseaux de neurones.
- *Aborder le problème de la conception en faisant abstraction de celui de l'adaptation.* C'est typiquement la démarche de l'IA symbolique classique : il s'agit de concevoir les robots à un niveau relativement abstrait (e.g. symbolique), comptant alors sur le développement des techniques adaptatives pour un jour résoudre le problème de la perception et de l'opérationnalisation effective des résultats obtenus à ce niveau abstrait.

B. LE CYCLE DE CONCEPTION

Nous examinons dans cette section comment le cycle de conception, et plus particulièrement de programmation, est abordé en robotique. L'idée générale est de décomposer le problème (tâche) que doit résoudre le programme en sous-problèmes (sous-tâches) indépendants. Nous connaissons deux façons de mener cette décomposition : l'analyse fonctionnelle descendante, qui décompose le problème en une hiérarchie de “niveaux de description”, et l'analyse comportementale, qui préfère le décomposer comme la collaboration de fonctionnalités parallèles et indépendantes.

L'analyse descendante constituera le §B.1, l'analyse comportementale le §B.2, et enfin le §B.3 décrira la notion de “modularité”, qui est commune à ces deux approches (et, plus généralement, de toute décomposition d'un problème en sous-problèmes).

Cette section et les suivantes s'appuieront très souvent sur un exemple commun : le problème de navigation décrit figure 1.[2]

[2] On cherche bien sûr à résoudre ce problème de la figure 1 par une méthode *automatique.* Une solution non automatique serait par exemple de munir le robot d'une télécommande et de laisser un opérateur humain guider le mouvement, ce qui est à écarter pour la robotique autonome.

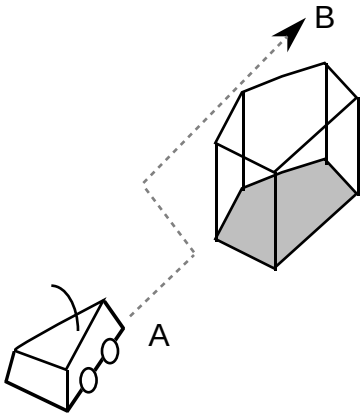


Figure 1. Un problème de navigation : le robot doit se rendre du point A au point B sans collision avec les obstables. La trajectoire représentée n'est qu'un exemple de solution.

B.1. L'analyse descendante

Pour résoudre le problème de la figure 1, l'idée la plus répandue est de procéder en deux étapes : premièrement déterminer une trajectoire qu'il suffirait de faire suivre au robot pour arriver au but sans collision, et deuxièmement faire effectivement suivre cette trajectoire au robot.

Planification de trajectoire

La première étape pose le problème ainsi : parmi les multiples trajectoires que pourrait suivre le robot, il s'agit d'en trouver une qui réalise la tâche demandée : partir de la position initiale A pour aboutir à la position finale B, et cela sans collision.[3]

Toutes ces contraintes sont de nature géométrique. L'idée est alors d'utiliser un modèle géométrique de l'environnement où les obstacles, les murs et le robot sont modélisés par des polyèdres, les collisions étant alors modélisées en termes d'intersections de volumes.[4] Dans ce modèle, le problème de la recherche d'une trajectoire s'exprime formellement par :

```
     { Une position initiale : (0,0)
[1]  { Une position but : (-8,65)
     { Des contraintes sur la trajectoire (pas de collision)
```

et dans ce cadre formel on peut automatiser la recherche d'une solution (Lozano-Perez 1993).

Toutefois, la spécification [1] ne suffit pas à déterminer une solution unique. Pour parvenir à un problème bien posé, il faut lever l'indétermination, et pour cela introduire d'autres contraintes non requises dans le problème d'origine. Par exemple, on peut chercher une trajectoire de longueur minimale, ou bien choisir aléatoirement les paramètres indéterminés, ou encore guider la décision par des critères heuristiques orientés vers une facilité d'implémentation informatique.

---

[3] Plus éventuellement des contraintes cinématiques de type non-holonomie. Dans l'exemple présent, le robot est considéré holonome.

[4] Plus précisément, il s'agit d'un ensemble d'équations et inéquations semi-algébriques (Lozano-Perez & al. 1993). La manipulation de tels modèle est appelée “géométrie algorithmique”.

Ces choix ayant été pris, on obtient par exemple comme solution la trajectoire représentée en pointillé dans la figure 1, et qu'on décrit comme une successions de points :

```
[2]  { (0,0) → (0,20) → (-8,25) → (-8,65)
       Les portions de trajectoires entre deux points étant linéaires
```

Cette phase de résolution [1]→[2] (rechercher une trajectoire) s'appelle classiquement “planification de trajectoire” (Latombe 1990).

Suivi de trajectoire

Passons à présent à la seconde étape : ayant obtenu une trajectoire, il s'agit de spécifier les mouvements du robot qui lui fasse suivre cette trajectoire.

La trajectoire ci-dessus est une succession de translations pures et de rotations pures, que l'on peut représenter de façon équivalente par le programme [2’] :

```
[2’] { translation (20cm); rotation (60°);
       translation (8cm); rotation (-60°);
       translation (40cm) }
```

Nous considérons que ce programme [2’] n'est toujours pas une solution au problème, mais que la solution opérationnelle est celle qui s'exprime en termes de commandes motrices.[5] Toutefois, la traduction ne semble guère poser de problèmes. Appelons `Δroues(x,y)` la commande motrice d'un incrément de `x` tours pour la roue gauche et de `y` tours pour la roue droite. La circonférence des roues étant de 16 centimètres, le programme [2’] se traduit par le programme moteur [3] :

```
[3]  { Δroues (+1.25,+1.25); Δroues (-0.8,+0.8);
       Δroues (+0.5,+0.5); Δroues (+0.8,-0.8);
       Δroues (+2.5,+2.5) }
```

et ce programme [3] semble bien résoudre le problème initialement posé. Cette phase de résolution [2]→[3] (suivre une trajectoire donnée) s'appelle “suivi de trajectoire” (Latombe 1990).

Cet exemple a fait apparaître trois niveaux de description :

- Le niveau “tâche” modélise la tâche de navigation, typiquement dans un modèle géométrique. On trouve parfois l'appellation “programme niveau tâche” pour la spécification [1].
- Le niveau “effecteur” modélise une trajectoire, typiquement par des transformations de repères cartésiens (langage LM, Mazer & Miribel 1985).
- Le niveau “moteur” exprime la solution recherchée en termes moteurs, ici la commande `Δroues(x,y)`. Le programme [3] est parfois appelé “programme moteur”.

La planification de trajectoire est le processus qui permet de passer d'une description niveau tâche à une description niveau effecteur, et le suivi de trajectoire est le processus qui permet de passer de cette dernière à une description motrice.

---

[5] C'est-à-dire comme une consigne sur les asservissements moteurs. Ce choix est en fait un peu arbitraire. Même analysé en termes de consignes motrices, le contrôle d'un robot reste très complexe et fait appel à de nombreux autres niveaux de description. Il est difficile de décréter à partir de quand un programme sera réellement opérationnel. C'est donc dans un souci de simplification que nous fixerons la limite au niveau de la commande, et considérerons ce qui relève de l'automatique comme idéalement réalisé.

B.2. L'analyse comportementale

Il existe une autre façon d'analyser le problème de navigation ci-dessus. Elle consiste à distinguer les deux tâches que doit accomplir le robot : éviter les obstacles et se rendre au but. L'idée est alors de programmer indépendemment ces deux tâches, pour qu'elle puissent se dérouler en parallèle, et de les combiner intelligemment pour que de leur coopération résulte le mouvement global du robot souhaité (Brooks 1986, Braitenberg 1983). De telles fonctionnalités sont appelées traditionnellement des “comportements” (behaviours).

Pour cela nous allons programmer chaque comportement comme le calcul en continu de commandes motrices en fonction de données sensorielles, ce que nous appelerons un “réflexe”.[6] Cela exige toutefois que les obstacles et le but soient repérables par des capteurs : nous considèrerons donc le robot muni d'un anneau de proximètres pour détecter les obstacles, et d'un anneau de cellules photoélectriques pour détecter le but, qu'on matérialise alors par une lampe (par exemple Pichon (1991) et Blanès (1991) présentent un robot similaire, mais avec une détection d'obstacles basée sur un principe différent).

Le comportement “aller au but” est implémenté en calculant les commandes `Δroues` à chaque instant comme une fonction f des cellules (un réflexe), et l'évitement d'obstacles est implémenté en calculant ces mêmes commandes comme une fonction g des proximètres (un second réflexe). Les deux réflexes se partageant les ressources motrices du robot, les exécuter en parallèle néc

essite de composer ces commandes en une commande effective par une opération $\oplus$ (typiquement une combinaison linéaire des contributions de chaque réflexe sur chaque commande). Le programme final est donc de la forme :

[4] `Δroues (`f`(cellules)` $\oplus$ `g(proximètres))` [les fonctions f et g retournant deux valeurs]

B.3. La notion de modularité

L'idée des démarches tant descendantes que comportementales est finalement la même : décomposer un problème en sous-problèmes *indépendants.* Chaque sous-problème (niveau de description ou comportement) peut alors être étudié, développé et testé localement comme un “module” séparé des autres, le système complet étant ensuite réalisé en programmant les interfaces et les interactions entre ces modules. Cette facilité de conception, très recherchée, est appelée “modularité”.

La modularité dans l'analyse descendante

Dans l'exemple décrit plus haut, les problèmes de planification et de suivi de trajectoire semblent typiquement pouvoir s'aborder indépendemment. La façon dont une trajectoire a été trouvée importe peu lorsqu'on se demande seulement comment la suivre, et réciproquement la façon dont on s'assurera du suivi d'une trajectoire donnée importe peu lorsqu'on se demande seulement si cette trajectoire va bien du point A au point B sans collision.

---

[6] Notons bien la différence faite ici entre un comportement (fonctionnalité du robot) et un réflexe (implémentation d'un comportement). Dans la littérature, il y a une ambiguïté fréquente sur le mot “comportement” (behaviour) qui peut désigner indistinctement les deux notions.

Cependant, il est clair que la solution à un problème de navigation peut être influencée par de nombreux facteurs de nature non géométrique, tels que les couleurs, matières, textures, comportements et historique des objets de la scène. Prendre en compte tous ces facteurs en bloc serait effroyablement complexe, mais le problème semble devenir gérable si on peut le décrire en termes exclusivement géométrique (volumes, trajectoires, collisions), parce que dans ce cas on sait pouvoir disposer des techniques de la géométrie algorithmique. Pour que cette démarche soit valable, il faut postuler que les aspects non géométriques de la situation pourront être gérés séparément à d'autres niveaux de description, ou bien pourront rester ignorés.

La notion hiérarchique de “niveau” est ici similaire à celle qui, dans les sciences physiques ou naturelles, lie des phénomènes micro- et macroscopiques tels qu'une distribution de vitesses de molécules gazeuses et une pression. De même que dans un tel système toutes les informations disponibles au niveau moléculaire deviennent non pertinentes dès qu'on les a résumées par une information de niveau macroscopique ; de même pour un robot toutes les informations de détail au niveau des textures, couleurs, etc… deviennent non pertinentes dès qu'on les a résumées par une information géométrique.

On cherche en somme à distinguer, parmi les connaissances innombrables potentiellement à considérer, un nombre restreint qui suffise à faire rouler un robot en ignorant les autres, la prise en compte de ces autres étant récursivement abordée par la même démarche réductrice — ou bien ignorée.

La modularité dans l'analyse comportementale

Dans l'analyse comportementale, l'idée de fond est la même : il s'agit de rendre abordable un problème complexe en concevant ses différents aspects comme autant de comportements indépendants. Pour l'analyse descendante la coopération des différents niveaux de description était réalisée par leur structuration hiérarchique explicite ; pour l'analyse comportementale la coopération des comportement est réalisée par la façon explicite de combiner les réflexes qui les implémentent.

La notion d'indépendance des niveaux de description devient ici une notion d'indépendance des comportements. Par exemple, il est vrai que la façon dont le robot évite un obstacle peut dépendre de la position du but et de tout ce que le robot est en train de faire “par ailleurs”. Cependant, si gérer tous ces aspects en bloc est trop complexe, on postule alors que l'aspect “évitement d'obstacle” peut être décrit séparément des autres aspects (tels que l'orientation vers le but), et qu'il suffira de bien composer ces comportement pour obtenir une solution au problème complet de navigation.

Ce postulat apparaît le plus clairement dans les travaux sur la “sélection des comportements” (action selection, behavioural choice : Maes 1991, Tyrrell 1992, Ring 1992). L'idée de base dans ces travaux est de considérer les comportements comme des problèmes séparés et en compétition[7] vis-à-vis des ressources disponibles. La composition et la coopération de ces comportements devient alors globalement un problème de séquencement :

---

[7] Le moteur de résolution des conflits est en général un état interne du robot, assimilé à une “motivation”.

«[Generating and controlling behaviour]: this basic task of animal brain can be split into three subtasks :

1- Sensing of the environment so as to be able to perceive what is going on at each moment in time (perception).

2- Taking the interpretation of the environmental situation and using it to decide which of the animal's repertoire of behaviours is the most appropriate (behavioural choice).

3- Transforming the chosen behaviour into a pattern of movements of parts of the body (motor control).» (Tyrrel & Mayhew 1990)

Dans les travaux sur la sélection de comportements, le point 3 ci-dessus est considéré résoluble indépendemment des deux premiers, et donc on s'autorise à l'imaginer résolu pour se focaliser sur les deux premiers.[8] On voit bien ici combien la notion de comportement devient aux approches comportementales ce que celle de niveau de description est aux approches descendantes.

C. CRITIQUE DU CYCLE DE CONCEPTION EN ROBOTIQUE AUTONOME

Nous voulons montrer dans cette section que les démarches de conception décrites ci-dessus posent de grandes difficultés en robotique autonome. La raison est que leur application concrète repose sur l'existence d'un concepteur humain, qui doit faire *systématiquement* usage de ses propres capacités d'adaptation. Or la particularité de la robotique autonome est qu'elle cherche justement à éliminer une partie de cette intervention systématique du concepteur.

Nous argumentons cela de la manière suivante :

Nous montrons d'abord que l'intervention du concepteur est nécessaire pour garantir que l'exécution d'un programme de robot mène bien au résultat souhaité. Nous constatons qu'en robotique cela n'est pas reconnu, et nous y voyons la marque d'une confusion entre abstrait et concret, c'est-à-dire entre une modélisation et sa mise en oeuvre. Nous appelons cela “le problème des conditions de validité” (§C.1).

Nous montrons ensuite que le concepteur humain est également le seul garant de la modularité effective de la décomposition d'un problème en sous-problème, et que cela non plus n'est guère reconnu en robotique (§C.2). Nous appelons cela “le problème de la modularité”.

Chacun de ces problèmes (conditions de validité et modularité) est illustré dans le cas de l'analyse descendante comme de l'analyse comportementale.

C.1. Le problème des conditions de validité

Analyse descendante

Reprenons, toujours sur le même problème de la figure 1, l'exemple du suivi de trajectoire. Il s'agissait, rappelons-le, de faire suivre au robot la trajectoire [2], sans se soucier de la façon dont elle avait pu être obtenue. La solution adoptée avait été le programme [3]. Mais il est clair que certaines conditions doivent être remplies pour que l'exécution du programme [3] ait effectivement pour résultat la trajectoire souhaitée [2] :

---

[8] On trouve sous l'étiquette “behaviour selection” des travaux comme Maes & Brooks (1990), qui au contraire se concentrent sur l'aspect moteur et échappent donc à cette analyse. Il s'agit pour nous de travaux de nature différente, qui concernent plus des réflexes que des comportements ; cf la note 6 précédente au sujet de l'ambiguïté fréquente dans l'emploi du mot “behaviour”.

Les deux roues sont en contact permanent et sans glissement avec le sol,
Le robot se situe bien à la position initiale au début de son mouvement,
Le sol est plan et horizontal,
Le robot ne rencontre effectivement pas d'obstacle (ni de trous) sur son chemin,
Les roues ont bien chacune une circonférence invariante de 16 centimètres,
Les axes de ces roues sont bien alignés,
Les asservissement commandés par `Δroues` fonctionnent correctement,
Etc…

Ces conditions portent sur des phénomènes qui n'ont été modélisés ni au niveau tâche, ni au niveau effecteur : glissements, incertitudes, aspect tridimensionnel du sol. En fait, à chaque niveau de description on avait postulé que les aspects non pris en compte le seraient à des niveaux “inférieurs” — mais on est arrivé en pratique au niveau purement moteur sans avoir effectivement pris ces phénomènes en compte.

Le concepteur, lui, est bien conscient de l'existence de ces conditions, et de la nécessité de s'assurer qu'elles seront respectées s'il veut garantir que le programme [3] résoud bien le problème posé. Cette situation est générale : un concepteur humain est toujours à même d'imaginer (ou de découvrir empiriquement) de nombreuses circonstances pour lesquelles une modélisation donnée devient inadéquate, et toute modélisation s'accompagne donc de conditions de validité informelles telles que ci-dessus, limitées seulement par l'imagination.

Le concepteur a alors plusieurs possibilités :

a) Il peut garantir que toutes ces conditions seront automatiquement respectées dans l'environnement où le robot est amené à évoluer.

b) Il revoit son analyse pour prendre en compte une condition qu'il ne peut garantir, et ainsi de suite jusqu'à ce que les conditions restantes puissent être garanties comme en (a).

c) Il est conscient que certaines conditions peuvent ne pas être remplies, et que dans ce cas le modèle deviendra inexact. Néanmoins, il pense que ce modèle restera une approximation suffisante, au moins pendant un certain temps. Le concepteur ne revoit donc pas son analyse, mais doit prévoir régulièrement des mises à jour du modèles à partir d'informations sensorielles.

C'est à partir de là que la robotique autonome se distingue de la robotique classique.

En robotique classique, toute lattitude est laissée au concepteur de spécifier lui-même tout ce qui ne fait pas partie explicitement du cahier des charges (c'est même là une de ses tâches principales). Le problème tel que posé figure 1 ne précisait pas que le robot doit pouvoir glisser, et par défaut le concepteur comprend qu'il a le droit de faire en sorte que le robot ne glisse pas — et ce choix, en fait, fait partie de la solution au même titre que le programme du robot. C'est cela qui permet à la démarche (b) — donc un bouclage entre conception, expérimentation et spécification de l'environnement — de converger vers une solution du type (a) qui résoud le problème.

En robotique autonome, nous nous intéressons aux environnements physiques et naturels, et le concepteur ne peut plus contrôler l'environnement. Nous prétendons que dans ce cas, cet environnement est souvent trop complexe pour que la solution (b) puisse mener à une modélisation satisfaisante de type (a). Ce peut être possible dans des cas très particuliers, mais par choix nous nous intéressons ici au cas contraire, que nous pensons être le plus fréquent.[9]

Il reste alors la possibilité (c), mais elle nécessite un mécanisme pour mettre régulièrement le modèle à partir d'informations sensorielles. Si ce mécanisme est conçu selon la même démarche descendante, on retombe sur le même problème. C'est là que la notion de mécanismes adaptatifs semble s'imposer — mais nous y reviendrons.

Analyse comportementale

Le problème des conditions de validité se pose également dans l'analyse comportementale. Soit un comportement attendu du robot, que l'on a implémenté par un réflexe. Le comportement effectivement observé n'est pas toujours celui attendu ; selon la situation concrète du robot le même réflexe peut réaliser des comportements différents.

Par exemple, reprenons le réflexe [4], qui programmait un évitement d'obstacles à partir de capteurs proximétriques. Concrètement, des proximètres sont sensibles à la couleur. Si les obstacles blancs sont ainsi facilement contournables, des obstacles noirs et mats mèneront souvent à une collision. Il y a donc des conditions de validité sur la couleur des obstacles, ainsi d'ailleurs que sur leur forme ou leur température.

La confusion entre abstrait et concret

Nous pensons que fondamentalement, le problème des conditions de validité reflète une confusion entre ce qu'est une solution abstraite à un problème et ce qu'est une solution concrète :

- Automatiser la résolution d'un problème (tâche robotique) passe toujours par une modélisation. Modéliser un problème, c'est en premier lieu choisir les termes dans lesquels on représente ce problème. Les analyses descendante ou comportementale sont des méthodologies pour guider ce choix, c'est-à-dire en définitive des méthodologies de modélisation.
- Cette modélisation est toujours une simplification, une abstraction permettant de manipuler formellement *certains aspects* du problème concret plus riche ; mais les termes choisis ne suffisent jamais à le caractériser *complètement.*
- La mise en oeuvre concrète de ce modèle abstrait nécessite par conséquent toujours des capacités humaines : juger de la pertinence a-priori du modèle et de son applicabilité, gérer certains détails non modélisés, évaluer les résultats obtenus. On ne sait donc automatiser totalement que la résolution de problèmes abstraits.
- En robotique, cet aspect est largement méconnu.

---

[9] La position prétendant que tout environnement naturel est systématiquement analysable à un niveau de détail suffisant pour ne pas avoir à le contrôler tient pour nous d'une philosophie, dite “objectiviste”, que nous discuterons (et rejeterons) plus loin dans ce chapitre.

Outre son aspect synthétique, cette vision en termes d'abstrait-concret souligne bien qu'il s'agit réellement d'une difficulté *de fond,* de nature conceptuelle et non seulement théorique ou technique.

Analyse descendante

Malgré son importance quotidienne, la notion de condition de validité n'est que rarement explicitée, voire reconnue, en robotique. Typiquement, un programme de suivi de trajectoire sera présenté sous la forme [2'] : `translation(20cm) etc…`, et non sous la forme [3] : `Δroues(1.25,1.25) etc…`. Et ce faisant, on oublie que `translation(20cm)` ne signifie *pas* que le robot effectuera une translation de 20cm dans l'environnement, mais bien que les roues font chacune un tour et quart. Les conditions de validité restent implicites.

Voici ce qui se passe, plus généralement :

- L'environnement d'un robot est toujours présenté de façon abstraite (modélisation).
- Cette abstraction résume une grande variété d'environnements concrets.
- On parvient à résoudre le problème dans l'environnement abstrait.
- Éventuellement, la méthode trouvée est validée dans un environnement artificiel justement conçu pour correspondre au modèle.[10]
- On en déduit que la méthode serait applicable dans tous les environnements concrets représentés par l'abstraction.

C'est ce dernier point qui est fallacieux. On considère généralement que plus un problème est abstrait, plus sa solution est de portée générale ; mais ceci n'est vrai que parce qu'*en fait plus les détails concrets devront être résolus par le concepteur humain.* C'est cette liberté dans l'application de la solution qui fait la généralité de celle-ci. Mais si l'intermédiaire humain est exclu, alors les aspects non modélisés, loins d'être laissés à son appréciation, se transforment en hypothèses implicites. Or, plus le problème posé est abstrait, plus il y a d'aspects non modélisés et donc d'hypothèses implicites, qui *restreignent* les situations gérables par le modèle sans intervention du concepteur.

Dans la pratique, les conditions de validité concernent toujours des caractéristiques importantes de l'environnement. On est continuellement confronté aux détails qu'on voulait justement pouvoir ignorer par le choix d'un modèle adéquat : calibrage, approximations obligées, données bruitées ou aberrantes, événements imprévus dans la scène. En fin de compte, les modèles très spécifiques sont au moins applicables aux problèmes concrets pour lesquels ils ont été conçus (e.g. Takeyasu 1977 et Kashioka 1977), tandis que les modèles qui se veulent très généraux ne sont applicables à aucun problème concret, tant leurs conditions de validité deviennent contraignantes (e.g. HANDEY, Lozano-Pérez & al. 1993). En

---

[10] En IA et en robotique, c'est bien souvent le modèle qui façonne l'expérience sur laquelle il sera validé. Les pièces manipulées par les bras de robots, par exemple, sont souvent usinées pour correspondre à leur modèle. En ce sens, même certains environnements physiques peuvent être considérés comme des “mondes jouets” (toy-worlds).

robotique les approches descendantes n'ont de résultats satisfaisants que dans des environnements très bien contrôlés.

Analyse comportementale

Décomposer un comportement global en un ensemble de fonctionnalités exécutées en parallèle, c'est se concentrer sur certains aspects de ce comportement de façon à savoir les gérer ; il y a là aussi un souci de simplification et d'abstraction.

Reprenons par exemple l'idée de voir le problème de la figure 1 comme la coopération d'un comportement d'évitement et d'un comportement de déplacement orienté vers une source lumineuse. Beaucoup de travaux se contentent d'adopter ce niveau d'abstraction sans se préoccuper des mécanismes effectifs (e.g. réflexes) qui seront utilisés pour implémenter ces comportements sur un robot réel dans un environnement physique et naturel. Citons notamment les travaux déjà cités sur la “sélection des comportements”, cf. SAB 90 et SAB 92. Or, dans la pratique, une telle implémentation nécessite pour fonctionner des environnements bien particuliers, voir par exemple les conditions de fonctionnement du robot analogique de Blanes (1991) et Pichon (1991).

Il est vrai que le problème pratique est moins apparent pour les approches comportementales que pour les approches descendantes, parce qu'elles n'utilisent pas un modèle de l'environnement, mais plutôt un modèle de *l'interaction* entre le robot et son environnement. Cette façon de faire est plus robuste à l'erreur. Par exemple, une erreur ayant entraîné une collision dans un modèle géométrique amène une erreur permanente sur la localisation du robot et de l'obstacle éventuellement déplacé lors de la collision, tandis que dans un modèle comportemental cette erreur peut perturber le fonctionnement du robot un moment mais redevenir valide dès que les conditions sont à nouveau respectées. Le contrôle exact de l'environnement est donc moins crucial pour les approches comportementales.

Cela ne doit cependant pas cacher que le problème de fond reste le même. La robustesse à l'erreur est un progrès, mais ne fait que reculer un peu l'attaque du problème : si on veut concevoir un robot ayant une certaine richesse de comportement dans un environnement physique et naturel (e.g. effectuer en parallèle une dizaine de tâches interdépendantes), et si on veut que bien que robuste aux erreurs le robot passe la plus grande partie de son temps à faire ce qui était prévu, cela demande une *analyse* détaillée, et dont la complexité ne nous semble pas moindre que celle d'une analyse descendante.

Pourquoi cette confusion ?

L'abstraction est donc, dans tous les cas, un outil puissant mais un *outil* seulement, dont la bonne utilisation n'est pas automatique mais passe par le savoir-faire du concepteur. Si cela n'est guère reconnu, nous pensons que c'est parce que nous sommes habitués à *poser* les problèmes de façon abstraite. Or, devant un problème abstrait, l'attitude habituellement adoptée est que tout ce qui n'a pas été explicitement spécifié doit être ignoré. Lorsqu'on n'a pas été mis en face de la situation réelle envisagée, et qu'on n'a donc pas la possibilité de s'y adapter en utilisant son propre “sens de la situation”, la convention est de formaliser le problème en fonction des seules données de l'énoncé — de ne pas introduire arbitrairement de

paramètres supplémentaires pour effectuer les calculs.[11] *En somme, on considère par défaut que celui qui a énoncé un problème abstrait a déjà effectué une partie essentielle de la formalisation en choisissant les paramètres pertinents à considérer.*

À notre avis, le piège de l'abstraction doit plus être évité dans la façon de poser un problème que dans la façon de le résoudre une fois posé, c'est-à-dire plus par une démarche conceptuelle et méthodologique que par une démarche théorique ou technique.

C.2. Le problème de la modularité

La modularité consiste, rappelons-le, à décomposer un problème en sous-problèmes résolubles a-priori séparément et indépendemment. Nous allons montrer que l'indépendance effective n'est pas acquise, et que c'est via l'intervention humaine sur l'environnement qu'elle est en général assurée.

Analyse descendante

Revenons à l'exemple de la planification de trajectoire. Il s'agit, rappelons-le, de passer d'une spécification niveau tâche [1] à une trajectoire [2] (niveau effecteur).

La planification est un problème *purement formel* : [1] et [2] s'expriment dans un même modèle formel, au sens où l'espace cartésien de [2] peut être formellement plongé dans le modèle géométrique de [1].

Entre deux niveaux formels, le problème des conditions de validité ne se pose théoriquement pas en, puisque toute l'information pertinente peut être explicitée. Le problème devient alors que l'indépendance des niveaux de description n'est en pratique jamais atteinte.

La raison à cela est la suivante. Nous avons déjà remarqué que la formalisation [1] du problème n'est pas assez contrainte pour déterminer une solution (trajectoire) unique. Il est nécessaire de fournir soit d'autres contraintes formelles (e.g. optimiser certains critères) soit des méthodes heuristiques (e.g. tirage aléatoire) pour lever l'indétermination. Ce choix est fait par le concepteur — et en pratique il sera déterminé en fonction des autres niveaux de description, brisant l'hypothèse d'indépendance entre niveaux.

Par exemple, la modélisation [1] n'impose rien sur la forme paramétrique a-priori de la trajectoire. Il faut faire un choix selon d'autres critères. On choisit en général de donner à la trajectoire une forme définie par morceaux, car cela permet une grande variété de formes à un moindre coût mathématique. Une trajectoire est donc décrite par une suite de points entre lesquels les portions de trajectoire ont une forme donnée. Celle-ci est en toute rigueur à déterminer en fonction du niveau inférieur, i.e. le suivi de trajectoire : il faudrait prévoir pour la planification les formes de trajectoires qui seront effectivement générées par le contrôleur du robot.

Dans la pratique cette solution est mathématiquement trop complexe. On préfère privilégier les formes sur lesquels les calculs de collisions sont simples, et choisir des portions de trajectoires linéaires : en fait le paramétrage [2] sera maintenu même si l'on sait que le contrôleur du robot est incapable de suivre des portions de trajectoires linéaires. Dans la réalité c'est très souvent le cas. Un robot mobile ne

---

[11] Cette convention est très clairement issue de notre culture scolaire.

peut en général pas changer arbitrairement de direction sur place (e.g. une voiture). De même, les trajectoires physiques d'un bras de robot sont en fait courbes (lorsque la planification se fait dans l'espace de travail).

Il apparaît alors une condition implicite qui est que la linéarité des portions de trajectoires décrites par [2] est une approximation suffisante des formes qui seront effectivement générées par le contrôleur du robot. Si la “condition de validité” ainsi réapparue n'est pas vérifiée, certaines trajectoires prévues sans collisions pourront résulter en une collision, et certaines solutions pourront ne pas être trouvées par le planificateur.

Que l'on choisisse la solution approchée ou la solution exacte, reste toujours le risque non chiffré qu'une modification au niveau inférieur entraine des répercussions importantes aux niveaux supérieurs. Contrairement au principe idéal de modularité, la “séparation” des niveaux de description n'est pas étanche.

Dans beaucoup de domaines d'ingénierie, cette difficulté est résolue par la compétence de l'ingénieur. Mais en robotique autonome, la fragilité de l'indépendance des niveaux de description est rendue problématique par le fait qu'au niveau le plus bas de la hiérarchie se pose systématiquement le problème des conditions de validité. Il semble en effet hasardeux d'affirmer que l'on peut traiter un problème à un niveau donné en faisant abstraction des niveaux inférieurs, en postulant que ceux-ci pourront être résolus dans un second temps seulement sans répercussion profonde sur les niveaux abordés en premier.

Conclusion : la modularité de découle pas automatiquement de l'analyse descendante ; et dans le cas de la robotique autonome elle semble très difficile à obtenir de cette façon.

Analyse comportementale

La notion de modularité pose aussi un problème dans l'approche comportementale en robotique. Programmer un robot par une collaboration de comportements indépendants demande que ces comportements soient eux-mêmes implémentés par des réflexes indépendants, ce qui n'est possible qu'avec une conception matérielle soigneusement adaptée.

Reprenons l'exemple du programme réflexe [4]. Nous avons imaginé ce dernier comme la collaboration de “éviter les obstacles”, comportement implémenté par un réflexe sur les proximètres seuls, et “rejoindre le but”, implémenté par un réflexe sur les cellules photoélectriques seules. Or, si nous nous rappelons l'histoire du KitBorg, il est clair que ces réflexes ont des interactions complexes via les ombres projetées par les obstacles, et que la programmation indépendante des deux comportements ne peut se faire que si les cellules photoéléctriques sont placées assez haut pour éviter les interférences avec les ombres (c'est d'ailleurs la solution adoptée par le robot mobile de Pichon et Blanes).

Ce genre d'approche nécessite donc une conception matérielle telle que les sources sensorielles soient utilisables indépendemment et sans interaction autre que via l'action du robot. Un bon exemple reste le contrôleur du bras du robot “Herbert” du MIT, décrit par Connell (1988), qui exploite l'idée de concevoir des capteurs ayant des rôles bien séparés et bien spécifiques. Connell défend cette position ainsi :

- Cela évite de faire de la fusion explicite de capteurs, qui pose toujours des problèmes compliqués de calibration croisée des capteurs les uns par rapport aux autres.
- Cela simplifie le traitement de la modularité, qui ne dépend plus de la structure interne de l'implémentation, les modules ne se communiquant pas de résultats intermédiaires. La modularité provient uniquement du fait que ces modules implémentent des comportements qui coopérent *via leurs conséquences dans le monde.*
- Il ne semble pas y avoir de limite à la taille potentielle que peut atteindre un système ainsi conçu.

Les deux premiers arguments sont convaincants — mais nous ne croyons pas au troisième. Cette démarche est réalisable sans doute pour les robots du type Herbert, dont le comportement global reste très pauvre, mais le développement de robots plus riches risque d'être plus freiné qu'aidé par l'exigeance d'indépendance des comportements. Au delà d'une certaine complexité, le dispositif sensoriel devra être exploité au maximum, les capteurs devront être considérés comme des ressources génériques autant que spécifiques, et pour cela une certaine fusion explicite des sources sensorielles risque de redevenir nécessaire.

La notion de sélection de comportements en compétition, quant à elle, devient franchement problématique. Dans notre exemple, cela reviendrait à dire que le robot se dirige vers son but, mais qu'en présence d'un obstacle il bascule vers un comportement d'évitement (plus ou moins brutalement, cf Ring 1992), pour revenir à son comportement initial ensuite. Or il est clair qu'aller au but et éviter les obstacles ne sont pas “en compétition”, mais ne sont que deux des aspects d'un comportement global souhaité (qu'ils ne suffisent d'ailleurs pas à déterminer complètement). Ensuite, leurs conceptions ne seront sans doute pas indépendantes : par exemple, le côté par lequel l'obstacle devrait être évité serait certainement décidé en fonction de la direction du but.

Conclusion : la décomposition en réflexes indépendants est une solution assez limitée au problème de la modularité. Il est nécessaire de considérer les interactions entre réflexes.

D. LE CYCLE D'ADAPTATION : PRÉSENTATION ET CRITIQUE

Nous venons, dans la section prédédente, de reconnaître que le rôle de l'humain dans la conception ne se résume pas à la modélisation, mais reste présent dans les moindres détails de sa mise en oeuvre. Nous nous demandons à présent dans quelle mesure ces détails peuvent être résolus non pas dans le cycle de conception, mais dans celui d'adaptation. L'adaptation permettrait-elle d'automatiser la gestion des conditions de validité d'un modèle, ou d'assurer une certaine modularité ? Serait-ce une réponse au problème de l'application concrète d'une solution abstraite ?

Rappelons d'abord en quoi consiste l'adaptation : on laisse dans un modèle certains paramètres non identifiés, et on donne une procédure pour les identifier à partir de données expérimentales. L'identification de paramètres est un sujet important dans l'industrie et il existe de nombreuses méthodes d'identification bien développées (Richalet 1991), mais nous n'étudierons la question ici que dans l'optique de la robotique autonome.

Voici le plan de cette section. Au §D.1 nous constaterons qu'en robotique, on privilégie souvent l'étude d'une forme d'adaptation particulière : l'apprentissage. Nous définirons ce que nous entendons par apprentissage, et nous ferons nous aussi le choix de nous restreindre à cet aspect de l'adaptation. Enfin, nous détaillerons l'aspect qui dans la notion d'apprentissage est le plus mis en avant pour la justifier : les paramètres ne sont pas fournis a-priori mais proviennent de données expérimentales.

Nous montrerons alors en quoi cet aspect de l'apprentissage, selon nous, ne résoud pas le problèmes des conditions de validité (§D.2).

D.1. L'apprentissage

Dans la quasi totalité des cas les informations sensorielles disponibles sont d'une autre nature que les paramètres à identifier ; ces derniers doivent alors être inférés à partir de celles-là. Dans un modèle polyédrique les positions cartésiennes des sommets ne sont pas directement mesurables, il faut par exemple les inférer à partir d'images de caméra. Ou encore, si on veut connaître la distance d'un obstacle en ne disposant que de capteurs infrarouges (proximètres) ou à ultrasons (sonars), il faut inférer la distance à partir de ces données.

Il reste alors deux méthodes pour aborder cette identification par inférence :

- Les données expérimentales peuvent être toutes considérées en bloc, et l'identification se fait seulement ensuite, “off-line”, sur une grosse masse de données déjà accumulées. Les techniques d'analyse de données se rangent généralement dans ce cadre (Bouroche & Saporta 1980).
- Les données expérimentales sont intégrées itérativement, “on-line”, au fur et à mesure du comportement du robot. Les nouvelles données modifient un peu les paramètres via des calculs locaux. C'est le cas pour la plupart des techniques neuronales, par exemple la backprop sur les perceptrons multicouches (Khanna 1990).

C'est la seconde approche (identification itérative on-line) que nous appellerons “apprentissage”. Les réseaux de neurones constituent l'implémentation la plus populaire des techniques d'apprentissage. Les architectures utilisées peuvent être de plusieurs types (perceptrons multicouches, cartes topographiques, réseaux de prototypes) et les algorithmes qui s'y adaptent aussi (descente de gradient, apprentissage par renforcement, algorithme de Kohonen ; cf Khanna (1990) pour une revue). Cependant, dans notre optique, des techniques moins populaires en IA, telles que le filtre de Kalman ou les méthodes de gradient stochastique, sont aussi des techniques d'apprentissage, même si elles ne sont pas exprimées en termes de réseaux.

Si pour un robot industriel on peut concevoir que l'identification d'un modèle (réglage des paramètres) puisse être traitée hors du fonctionnement normal du robot, pour un robot autonome il est souhaitable que les paramètres puissent évoluer pendant son fonctionnement normal. C'est donc l'apprentissage qui nous intéresse le plus dans cette optique, et nous nous y restreindrons (par choix). L'identification off-line reste néanmoins une voie potentiellement intéressante, qui se déroulerait à une échelle de temps plus grande ; mais nous n'en parlerons pas ici.

L'identification automatique de paramètres par apprentissage présente des avantages certains par rapport à leur détermination analytique par le concepteur.

- Avantage technique : la mise en oeuvre d'une technique d'apprentissage pour résoudre un problème est souvent plus aisée que l'analyse exacte ce problème. Typiquement, pour contrôler en termes de repères cartésiens la pince d'un bras de robot dont la géométrie est connue, il est plus facile d'apprendre empiriquement la correspondance entre coordonnées articulaires et coordonnées cartésiennes que de résoudre les équations géométriques définissant analytiquement ce changement de coordonnées.
- Avantage conceptuel : il semble plus juste et plus fiable d'identifier des paramètres à partir de données réelles, mesurées expérimentalement, plutôt que de leur imposer des valeurs déterminée “par ailleurs” par le concepteur. Nous appellerons cela “l'aspect expérimental” de la notion d'apprentissage.

L'avantage technique est indiscutable, et nous en préciserons quelques raisons plus loin. L'avantage conceptuel que constitue l'aspect expérimental ci dessus, en revanche, ne suffit pas à résoudre le problèmes des conditions de validité, et c'est ce que nous allons montrer maintenant. Il ne faut pas voir cette critique comme une condamnation de la notion d'apprentissage : nous verrons dans la section suivante que cette notion revêt un autre aspect, selon nous plus important que l'aspect expérimental bien que beaucoup moins abordé dans la littérature sur l'apprentissage, à savoir la possibilité de gérer explicitement l'incertitude. Les paragraphes ci-dessous, eux, ne concernent que l'aspect expérimental, leur but étant de montrer que le seul fait d'utiliser des données expérimentales ne suffit pas à résoudre nos problèmes.

D.2. A-t-on résolu le problème des conditions de validité ?

Reprenons l'exemple des modèles polyédriques de la géométrie algorithmique. Être capable de reconstruire une scène polyédrique à partir d'images, par exemple, ne permettra jamais de tenir compte des aspects purement dynamiques de la scène ou des aspects liés à la courbure des objets (comme le fait qu'une balle puisse rouler), car ces aspects ne sont pas liés aux paramètres du modèle, mais à sa structure même, au choix que ce modèle dénote de ne considérer que certains aspects de la situation. Si alors le robot se trouve dans une situation pour laquelle le modèle est inadéquat, le fait que les paramètres du modèle aient été identifiés expérimentalement et non a-priori n'aide en rien à résoudre le problème des conditions de validité.

On pourrait objecter que cela provient de l'utilisation intermédiaire d'un modèle interne de l'environnement (le modèle polyédrique). Cherchons alors à apprendre un comportement du robot (quel qu'il soit) directement en termes de réflexes : les commandes motrices M sont calculées comme une fonction f de l'ensemble des capteurs C du robot, $M=f(C)$. La fonction f peut être approximée par des techniques d'apprentissage, par exemple un algorithme de rétropropagation du gradient sur une architecture de perceptron multicouche (qui constitue un “approximateur universel” de fonction, Hornik & al. 1989).

Pour qu'une telle démarche soit satisfaisante, toutefois, il faut que la relation entre C et M soit effectivement de nature fonctionnelle, c'est-à-dire que pour une valeur de

C fixée corresponde en gros une valeur de M fixée. Si les données ne vérifient pas cela, l'apprentissage ne donnera pas de résultats valables (i.e. prédictifs). Dans un environnement naturel, les phénomènes liés par des relations fonctionnelles sont rares, et une telle hypothèse constitue un acte de modélisation important. Le problème des conditions de validité apparaît dès que l'on veut assurer automatiquement que cette hypothèse de dépendance fonctionnelle restera vérifiée en pratique.

Devant ces difficultés, on raisonne parfois ainsi : il ne faut pas chercher à imposer a-priori les relations pertinentes, mais les laisser “émerger” d'une structure sous-jacente suffisament riche et complexe pour cela. Cet appel à la notion d'émergence nous semble toutefois un leurre, pour les raisons suivantes.

La notion d'émergence existe en gros sous deux formes (voir Bonabeau 1992 pour une revue détaillée) :

- La première est la plus générale : un comportement émergent est un comportement qui peut être observé sans être explicitement présent dans la programmation du robot. L'évitement d'obstacle dans l'histoire du KitBorg est un exemple de comportement émergent. Chez les animaux, la dérive clinocinétique[12] est un comportement émergent (Bovet & Benhamou 1985).
- La seconde est liée à la dynamique des systèmes complexes et à l'opposition entre phénomènes microscopiques et macroscopiques (Dumouchel & Dupuy 1981). Nous l'appellerons plutôt “auto-organisation” ; c'est pour nous un cas particulier d'émergence (“structures” émergentes, cas particulier de “comportements” émergents). Par exemple, l'établissement de chemins stables chez les fourmis (Deneubourg & Goss 1990) ou la réorganisation périodique de certaines ruches (Hogeweg & Hesper 1985) sont des comportements émergents par auto-organisation.

On retrouve ici la notion de niveaux de description, mais présentée “à l'envers” : un niveau donné n'est plus décrit en soi, *a-priori,* mais par référence aux mécanismes qui ont permis de le faire émerger. Une fois qu'il a effectivement émergé, on peut *a-posteriori* le décrire à un niveau fonctionnel faisant abstraction du mécanisme qui le réalise.

Cette position renversée nous semble saine, et a l'avantage de bien amener et formuler les problèmes. Mais pour le roboticien, elle n'est guère constructive. Pour concevoir la réalisation d'un comportement émergent, il n'y a pas de magie, deux attitude sont possibles :

- Partir d'une fonctionnalité souhaitée a-priori, et se demander comment trouver, explorer, évaluer, exploiter un mécanisme l'implémentant par émergence ;
- Partir de techniques connues avec lesquelles on sait réaliser telle ou telle fonctionnalité émergente, et trouver un problème permettant d'exploiter ces techniques.

---

[12] La clinocinèse est un mécanisme d'orientation basé sur des stimuli non directionnels. La sinuosité de la trajectoire varie selon le gradient spatial du stimulus (modèle décrit par Bovet, 1988), et cela résulte en une trajectoire qui semble dirigée vers un but avec une exploration stochastique optimale de l'espace (diffusion). Ces deux propriétés, qui peuvent être décrites comme des fonctions que remplit ce mode de locomotion, ne sont nulle part explicites dans le mécanisme lui-même, qui n'est en rien de nature directionnelle.

Les difficultés de la conception restent donc posées de façon tout à fait classique. La connotation de “surprise” ou d'imprévu, qui souvent accompagne la notion d'émergence, n'est pour la conception pas une aide mais un problème à aborder.

Néanmoins, l'émergence semble souvent apparaître comme une solution en elle-même, et non comme une problématique. Il suffirait de rendre une situation suffisamment complexe pour que finissent par émerger certains phénomènes qui seront justement, par un heureux hasard, la solution à nos problèmes — mais on se refuse à décrire comment et pourquoi ce seront ces phénomènes particuliers qui apparaîtront.[13]

Faire appel à la notion d'émergence comme une alternative à celle de modéle, c'est éluder la question de la conception. Pour nous, cette notion ouvre des possibilités et pose des questions, mais ne fournit pas d'indication sur la façon d'exprimer un problème, de trouver les termes permettant de le résoudre.

En conclusion de ce paragraphe, le problème des conditions de validité est lié à la structure du modèle plus qu'à la valeur de ses paramètres, et ce n'est donc pas l'apprentissage de ces paramètres qui peut aider à le résoudre. Ce n'est que lorsque le modèle est adéquat, et que donc le problème qui reste n'est que d'identifier les paramètres, que l'apprentissage peut aboutir.

E. NOTRE RÉPONSE AUX PROBLÈMES SOULEVÉS JUSQU'ICI

À ce point de notre réflexion, nous avons soulevé un certain nombre de problèmes particuliers à la robotique autonome, et dont la clef semble résider dans la nature du rôle de l'humain dans la construction d'un robot. C'est ce rôle que l'on cherche à minimiser en robotique autonome, alors que selon nous il est loin d'être bien compris et bien cerné. Dans cette section et la suivante, nous avançons plusieurs idées pour remédier à cela, passant donc, après des sections essentiellement critiques, à deux propositions plus “positives”.

La première idée, qui fait l'objet du §E.1, concerne le cycle d'adaptation (d'apprentissage, plus précisément). Nous venons d'affirmer que les problèmes des conditions de validité et de la modularité n'étaient pas résolus par l'apprentissage. En fait, pour être précis, nous avons montré que *l'aspect expérimental* de l'apprentissage ne nous aidait guère pour cela. Nous voulons dans cette section présenter un autre aspect de la notion d'apprentissage qui, lui, nous semble ouvrir des voies intéressantes pour résoudre le problème des conditions de validité : la possibilité d'une gestion explicite de l'*incertitude.* La plupart des techniques d'apprentissage n'exploitent pas cette possibilité, qui sera plus amplement développée aux chapitres 3 et 4.

La seconde idée, exposée au §E.2, est que le développement d'un robot en deux cycles, conception et adaptation, n'est pas adéquate. Il manque entre les deux un troisième cycle, que nous appelons “incrémentalité”. Nous détaillons alors la façon dont nous concevons l'incrémentalité, et nous montrons que si la gestion de l'incertitude est une réponse au problème des conditions de validité, l'incrémentalité quant à elle peut être vue comme une réponse au “problème de l'imprévu”.

---

[13] Citons la conclusion de Pierre Marchal à l'issue des journées NSI (dont le but est de rassembler ingénieurs et biologistes pour jeter des ponts entre les disciplines) en mai 1994, qui disait en substance : “Le conseil principal des biologistes aux ingénieurs, c'est de cesser d'attendre de la notion d'émergence qu'elle résolve les problèmes à votre place”…

E.1. La notion d'incertitude

Le problème des conditions de validité est lié à l'inadéquation d'un modèle plutôt qu'à la valeur de ses paramètres. Si le robot se trouve dans une situation où un modèle qu'il utilise est peu adéquat, et qu'il est incapable de remettre en cause expérimentalement les prédictions du modèle, le problème des conditions de validité se pose crucialement.

Par conséquent, nous estimons qu'*une façon de bien gérer ce problème serait de faire en sorte automatiquement que le modèle devienne explicitement moins prédictif dans les circonstances où il est moins adéquat. L'objectif que nous nous fixons désormais pour résoudre le problème des conditions de validité est alors le suivant : les données expérimentales doivent permettre de quantifier automatiquement le degré d'incertitude (la confiance) accordé(e) aux prédictions du modèle.*[14]

Le robot peut alors “reconnaître” de cette façon qu'un modèle se révèle peu performant, et c'est le cas quand certaines conditions de validité ne sont pas vérifiées. Il s'agit alors de faire automatiquement “au mieux” avec les connaissances dont le robot dispose, même si elles se sont révélées expérimentalement imparfaites. Comment peut-on faire ? C'est un problème supplémentaire qui survient par rapport aux techniques ne prenant pas en compte l'incertitude, mais qui nous semble plus abordable que celui des conditions de validité dans sa généralité.

Prenons l'exemple d'un réseau neuronal permettant d'approximer une relation fonctionnelle f entre des entrées x et des sorties y, relation paramétrée par l'ensemble w des poids du réseau : $y=f_w(x)$. Une fois les paramètres w identifiés, ce modèle est utilisé ainsi : une valeur donnée de x détermine une valeur donnée de y, toujours la même. Alors, s'il se trouve qu'en fait x ne détermine pas réellement y, ce modèle est inadéquat et la valeur obtenue pour y est difficilement exploitable.

Plutôt que chercher une relation fonctionnelle, nous voudrions plutôt savoir, pour une valeur donnée de x, quelles sont les valeurs de y qui peuvent convenir, *et dans quelle mesure.* Cela donne beaucoup plus d'information, et c'est ce que nous appelons “gérer l'incertitude”.[15] Si le modèle s'est révélé très juste expérimentalement, une valeur particulière de y sera hautement favorisée (l'incertitude déterminée expérimentalement sera faible) et on retombera dans le cas précédent. Si au contraire x ne se révèle que faiblement informatif sur y, alors une grande plage de valeurs de y seront proposée de façon à peu près équivalentes. On reconnaîtra ainsi que le modèle n'a pas la puissance prédictive que l'on pouvait en espérer, et qu'il reste une grande liberté de choix qu'il n'a pas suffit à restreindre. Les résultats peuvent toutefois rester intéressant, biaiser le choix vers certaines plages de valeur, *contribuer* à ce choix plutôt que le *déterminer.*

---

[14] La notion d'incertitude d'un modèle en tant que confiance que doit lui accorder le robot est à bien distinguer de la notion plus habituelle d'imprécision ou de “bruit”. L'incertitude est un effet de l'ignorance du robot. Les simplifications assumées par un modèle, l'ignorance de certaines conditions de validité, les approximations effectuées : tout cela rentre dans le cadre de l'incertitude. Le bruit quant à lui est souvent présenté comme dû à l'imprécision des calculs et à l'imperfection des capteurs, ce qui n'est qu'un cas très particulier d'incertitude.

[15] Il existe d'autres façon d'aborder l'incertitude, qui consistent à trouver des méthodes de manipulation “robustes” (i.e. réduisant les conditions de validité à assurer) pour effectuer certaines tâches précises (Mason 1989, Malcolm & Smithers 1990, Chaumette 1990). Ces approches sont plus orientées vers les “techniques de l'ingenieur” . Notre objectif est plutôt une *approche de principe pour gérer l'inadéquation des modèles* utilisés, afin d'aborder non seulement le problème des conditions de validité mais aussi celui de l'imprévu (voir plus loin).

Des exemples concrets de gestion explicite de l'incertitude ne pourront être donnés qu'au chapitre 4, aprés que nous ayons exposé au chapitre 3 une théorie de l'incertitude. Nous reportons donc à plus tard les développements concrets de cette notion d'incertitude.

Notons, pour finir, que la gestion de l'incertitude constitue une réponse “négative” au problème des conditions de validité (ne pas faire entièrement confiance à tel ou tel modèle) plus qu'une réponse “constructive” (donner les indices pour trouver un modèle adéquat). Tenir compte de l'incertitude, c'est-à-dire faire au mieux avec et malgré les connaissances imparfaites dont on dispose, ne permet pas de résoudre le problème de “l'incomplétude”, c'est-à-dire découvrir les connaissances qui manquent ainsi que le moyen de les prendre en compte. Pour cette raison, nous ne pensons pas que l'approfondissement des techniques d'apprentissage puisse fournir en retour un bon guide pour aborder le problème de la conception plus tard. Plutôt que séparés hiérarchiquement, les deux cycles (conception et adaptation) sont parallèles.

Cette remarque permet de faire le lien avec la seconde idée que nous voulons exposer dans cette section, celle d'un cycle incrémental. Ce cycle est justement pour nous une façon d'aborder le problème de l'incomplétude, et constitue le “chaînon manquant” entre conception et adaptation.

E.2. Le cycle de l'incrémentalité

Dans la pratique, le développement d'un robot ne suit pas le schéma de principe conception-adaptation décrit jusqu'ici. Il y a toujours un retour de l'expérimentation sur la conception qui fait que le robot est re-conçu cycliquement. Nous dirions même que souvent, le problème de l'ingénieur, plus que de trouver une solution du premier coup, est de mener efficacement ce type de processus de mise au point pour converger vite vers une solution correcte. En informatique, nous pouvons appeler cela “débugger” ou “mettre au point” un programme.

Nous aimerions ici ne plus voir ce processus comme une simple mise au point, mais l'aborder comme un cycle de développement ayant son rôle à part entière, et que nous nommerons “incrémentalité”. Nous considérons que puisque *dans la pratique* les gestions de la conception et de l'adaptation ne sont pas séparés, la démarche réductrice qui cherche à les maintenir séparés doit être remise en cause, et le cycle incrémental est l'alternative que nous proposons.

L'idée de l'incrémentalité est la suivante. Dans le développement d'un robot en terme de conception et adaptation, le concepteur humain intervient sur deux tableaux :

- la conception du robot proprement dite (structure physique et logicielle, mécanismes adaptatifs…),
- la spécification et le contrôle de l'environnement.

Nous cherchons alors à éliminer l'intervention sur l'environnement en la remplaçant par une re-conception du robot. Cette re-conception, toutefois, ne doit pas concerner toute la structure du robot, car dans ce cas nous sommes à nouveau confrontés au problème entier de la conception, et nous ne disposons d'aucun guide systématique pour nous aider dans cette re-conception globale. Au contraire, étant donné un robot qui concrètement se révèle non satisfaisant car son bon

fonctionnement nécessite une intervention inacceptable sur l'environnement, *l'incrémentalité consiste à enrichir la structure du robot en conservant le plus possible de la structure existante et des identifications déjà effectuées.* C'est en cela qu'il s'agit d'un cycle de développement intermédiaire entre les deux autres : d'une part il s'agit de modifier la structure (alors que l'adaptation est purement paramétrique), mais d'autre part il s'agit de conserver une partie de cette structure (alors que la conception consiste à construire une nouvelle version du robot, à ré-analyser le problème d'une façon non systématisable).

Ultimement, nous cherchons à automatiser ce processus incrémental. Toutefois, pour l'instant, nous ne présentons l'incrémentalité que comme une méthodologie, destinée au concepteur. Cela conduit, paradoxalement, à *redonner* dans un premier temps un rôle important au concepteur. Le paradoxe s'explique par le fait qu'avant de chercher à en automatiser une partie, *nous en sommes actuellement encore à reconnaître ce rôle et à l'analyser.*

La notion d'incrémentalité a déjà été abordée en robotique, mais par une optique différente de la nôtre, que nous décrivons ici avant de revenir à notre propre conception du cycle incrémental.

L'incrémentalité de Brooks

Le précurseur de la démarche incrémentale en robotique est Braitenberg (1983), qui décrit une série de robots de plus en plus sophistiqués, chaque étape dépendant du bon fonctionnement des robots des étapes précédentes, ainsi que de l'expérience (la compréhension des phénomènes mis en jeu) que le concepteur aura retirée de leur développement. Braitenberg montre bien la richesse potentielle de cette approche, mais les expérimentations y sont faites “par la pensée” (thought experiment). En robotique réelle, l'approche de référence en matière d'incrémentalité est celle de Brooks.

Brooks a le premier appliqué cette démarche à la robotique, et l'a popularisée par la “subsumption architecture” (1986, 1989; Connell 1988). Cette architecture permet d'implémenter des réflexes sur des automates d'états finis asynchrones, la coopération entre ces réflexes étant réalisée par un système d'inhibition et de suppression entre automates.

L'idée initiale de Brooks, illustrée par la figure 2a, est de développer un système robotique par “couches” (layers) successives, chaque couche rajoutant une fonctionnalité (un comportement) au système. Par exemple, la première couche, `Avoid Object`, est un réflexe faisant réagir le robot à la proximité d'un objet (détecté par un anneau de sonars) : le robot s'écarte lorsqu'un objet s'en approche, mais reste autrement immobile. Une fois cette couche bien mise au point, on va en ajouter une seconde, `Wander`, grâce à laquelle le robot se déplace même en l'absence d'obstacles, explorant son environnement aléatoirement. La coopération entre les deux couches est réalisée par la possibilité pour une couche d'en inhiber une autre ou de la remplacer. En l'absence d'obstacles, la couche `Wander` commande le robot, mais lorsqu'un obstacle est détecté `Avoid Object` reprend la main. Une fois le comportement global satisfaisant, on attaque la réalisation d'une troisième couche pour diriger l'exploration vers un but lorsque les conditions favorables sont détectées, par exemple se diriger vers les portes ou suivre les couloirs ; et la démarche se poursuit ainsi vers des comportements de plus en plus riches (figure 2b).

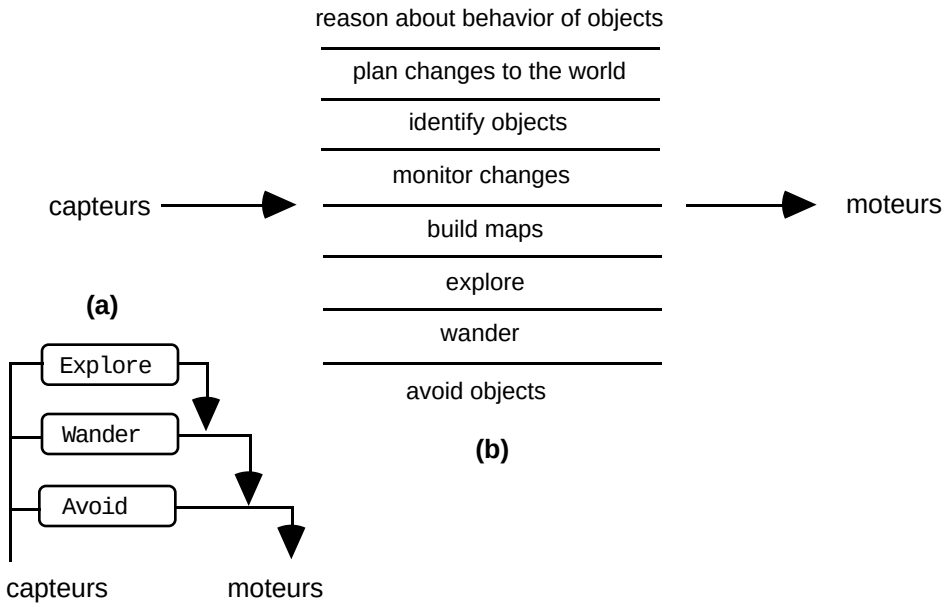


Figure 2. La conception “par couches” de Brooks. (a) On commence par mettre au point certaines fonctionnalités de base (évitement d'obstacles), puis on vient en greffer de nouvelles (exploration aléatoire, puis dirigée) qui exploitent à chaque fois les fonctionnalités existantes, par inhibition ou remplacement des signaux contrôlant celles-ci. (b) Les huit couches (fonctionnalités) imaginées comme exemple par Brooks (1986).

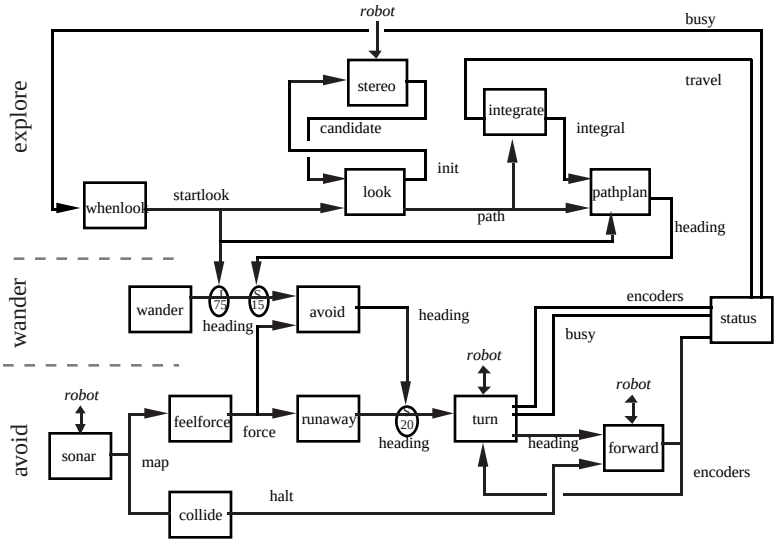


Figure 3. Le détail de l'implémentation des couches `Avoid`, `Wander` et `Explore`, reproduit d'après Brooks (1991). La légende d'origine dit “lower level layers never rely on the existence of higher level layers”, mais il semble que si c'est vrai concernant le *fonctionnement* d'une couche (layer), sa *conception,* elle, dépend en partie des couches supérieures. Par exemple, le niveau de la boîte “status” n'est pas clair : le signal “travel” n'est nécéssaire que pour `Explore`, mais la structure de la couche `Avoid` est clairement influencée par “status”. De même, le signal “force”, entre les boîtes “feelforce” et “runaway” de la couche `Avoid`, semble avoir dû être explicité parce qu'il s'est révélé nécessaire à la couche `Wander`.

Développer un système par étapes, en se fondant à chaque étape sur les capacités déjà implémentées, s'est révélé difficilement praticable. Il faudrait que ces capacités soient exploitables pour la suite du développement sans subir de modification. En pratique, on constate qu'à chaque étape il faut souvent tenir compte du moindre détail d'implémentation des capacités existantes, et que des modifications rétrospectives sont inévitables à tous niveaux. Par exemple, les trois premières couches construites par Brooks pour le robot “Allen” sont détaillées figure 3. La conception apparaît clairement globale plutôt que par étapes, au sens que chaque étape nécessite une reconception globale du système plutôt qu'un simple ajout de compétences sur le système existant. Quant à développer incrémentalement une collection de fonctionnalités complexes et interdépendantes (comme celle de la figure 2b), chacune reposant sur la précédente, cela nous paraît une gageure.

C'est ce défaut que Connell (1988) comptait aborder par l'idée de “communication via l'environnement” :

> «… it was quite common for a higher level behavior to tap off the outputs of certain modules in a lower level. The reason we eschew this practice is for the sake of modularity. There may be a number of ways to implement a particular behavior; however, only a small number of these will have the correct internal structure to allow the higher level to operate. Thus we may get a viable debugged low-level behavior only to find that we have to radically restructure it so that the appropriate internal signals are available to higher levels. This seems to defeat the claim of incremental extensibility. …
>
> «For these reasons, we have adopted the maxim “let the behavior be the interface”. That is, do not depend on the internal structure of some behavior, depend, rather, on this behavior's effect on the world. This way, as long as the behavior achieves the same task, it does not matter how it happens to be implemented.»

Grâce à ce choix[16], l'équipe de Brooks a atteint des résultats remarquables en robotique autonome. Nous avons cependant déjà expliqué (§C.2) que la modularité proposée ci-dessus nous semblait problématique pour réaliser des systèmes un peu complexes. Actuellement, ce sont plutôt des travaux comme ceux de Horswill (1993) qui semblent le plus refléter l'avenir de cette approche : il s'agit d'analyser en profondeur certains environnements très spécifiques pour proposer des réflexes adéquats dans ces environnements. On s'oriente donc vers l'élaboration de “catalogues de réflexes”, destinés aux concepteurs humains (une des questions abordées par Horswill étant d'ailleurs la recherche de principes taxonomiques pour un tel catalogue). Cette démarche s'est résolument tournée vers l'ingénieur et vers la spécification des environnements.

L'incrémentalité par la représentation

Nous avons pour notre part choisi une autre voie que Brooks pour attaquer le problème de l'incrémentalité, qui est de retourner à la notion de représentation. Nous espérons ainsi pousser plus loin la réflexion sur la nature du processus incrémental, pour plus tard espérer en automatiser une partie. Cette voie n'est pas concurrente de la précédente puisque les préoccupations ne sont plus vraiment les mêmes.

---

[16] Nous considérons en effet que c'est ce choix qui est fondamental, et non spécialement l'architecture subsumption elle-même, qui n'est pour nous qu'un langage de programmation.

La notion de représentation est en général très critiquée par les approches incrémentales. La position de Brooks sur la notion de représentation, largement répandue, est que “le monde est sa propre représentation”. On entend par là que les propriétés du monde prises en compte par le robot ne sont pas mesurées sur un modèle intermédiaire, mais directement via des capteurs. Il s'agit d'un principe méthodologique : les réflexes doivent “communiquer par le monde” plutôt que par le partage de résultats intermédiaires internes.

Pour nous, cette position est ambiguë dans la mesure où ce n'est pas le monde qui est sa propre représentation, mais bien les données provenenant des capteurs, et que ces capteurs ont été justement choisis spécifiquement parce que le concepteur avait à l'avance une idée de la façon dont il allait les utiliser, selon en fin de compte sa propre “conception du monde”. La position que “nous n'utilisons pas de représentation” est à notre avis une illusion rendue possible et fructueuse par le choix de la méthodologie particulière de Brooks (réflexes séparés et indépendants), mais elle n'est en rien intrinsèquement nécessaire à l'incrémentalité. Tentons donc de reconsidérer cette position.

Quand on veut réviser la structure d'un robot sans toucher à certains acquis, il faut préciser ce qui, dans la structure d'un robot, peut être amené à changer par incrémentalité, et ce qui au contraire doit rester invariant pour ne pas perdre dans ces changements toute l'expérience passée que l'on souhaite encore exploiter. La notion de représentation est pour nous le moyen de faire cette distinction, de désigner ce qui doit rester constant et pour quelle raison. Une représentation interne, c'est alors une trace de l'expérience passée qui soit *génériquement* exploitable, et qui doit donc être conservée. Ce sont ces préoccupations qui motivent le retour à la notion de représentation.

L'incrémentalité comme une réponse au “problème de l'imprévu”

Nous avions dans l'introduction décrit le “syndrome des pas de portes” ; il s'agissait rappelons-le de l'influence inattendue d'un pas de porte en cuivre sur le mouvement d'un robot. Dans l'anecdote décrite, le résultat était d'empêcher le robot de réaliser la tâche désirée. Ceci n'illustre en soi que le problème des conditions de validité. Le problème de l'imprévu[17] se pose lorsqu'on se demande : “qu'est-ce-qu'on fait maintenant ? comment désormais tenir compte de ce phénomène jusqu'ici imprévu ?”… Le problème de l'imprévu ne concerne donc pas les performances d'un robot tel qu'il est donné à un certain moment, mais le développement de ce robot, la façon de modifier après-coup sa structure pour tenir compte d'un imprévu.

Que peut-on faire concrètement face au syndrome des pas de portes ? Nous allons d'abord présenter certaines solutions classiques, puis d'autres solutions plus inhabituelles. Nous montrerons alors que ce sont ces dernières qui sont rendues possibles par l'incrémentalité, et qu'elles ouvrent des possibilités intéressantes.

Les solutions classiques sont les suivantes :

---

[17] En français, le verbe “prévoir” peut avoir plusieurs sens. Il peut être synonyme de “prédire”, ou bien signifier “envisager, prendre en compte”. C'est dans ce dernier sens que nous l'entendrons ici ; ainsi “imprévu” signifie non pas “surprenant” mais “non pris en compte”. On peut avoir pris en compte (prévu) un événement très improbable ; et on peut réciproquement prédire un événement que l'on n'a pas pris en compte (imprévu), et par exemple essayer, du coup, de le prévoir s'il est encore temps.

- On démonte le pas de porte pendant les expérimentations ou les démos. Cela revient à transformer l'environnement pour qu'il soit conforme au modèle.
- On assume que les pas de portes, s'ils ne sont pas à proprement parler des éléments anormaux de l'environnement, sont suffisamment rares pour que le fait qu'ils “plantent” plus ou moins le robot soit acceptable. Après tout, il arrive même aux êtres vivants d'avoir des accidents, le principal est qu'ils soient rares et ne remettent pas en cause leur viabilité générale.
- On met du cahoutchouc plus adhérent autour des roues du robot pour le rendre insensible aux glissement de ce type. Alors les pas de portes n'ont plus de conséquences sur le comportement du robot, et on peut continuer à les ignorer.
- On modifie le modèle (la carte du laboratoire) pour prendre en compte les pas de portes. On considère que c'est un élément initialement oublié dans la conception, mais qu'il faut savoir gérer maintenant qu'on a découvert son importance. Par exemple, on en fait un endroit où le robot devra systématiquement s'arrêter pour vérifier et corriger sa position sur la carte à partir d'informations extérieures (sensorielles), ou encore demander à l'opérateur de le replacer correctement.
- On abandonne la génération automatique de trajectoire pour une programmation réflexe ou une commande référencée capteurs (Chaumette 1990), par exemple un asservissement à un gyroscope embarqué pour assurer la droiture de la trajectoire. On a alors reconnu que le problème du glissement, bien qu'associé aux pas de portes puisque c'est cette situation particulière qui l'a mis en évidence, est plus générique que cela et remet en question la conception initiale.

Ces solutions devant l'imprévu peuvent se résumer comme un compromis entre deux critères :

- *Fréquence* : le phénomène peut être plus ou moins “normal” dans les situations où le robot est amené à fonctionner. Les pas de portes sont fréquents dans les immeubles et il serait gênant qu'un robot patrouilleur générique ne puisse pas les gérer, mais la présence d'une piscine intérieure est suffisamment rare pour rester hors du domaine de fonctionnement d'un tel robot.
- *Interférence* : le phénomène imprévu peut interférer avec les fonctionnalités du robot de façon plus ou moins importante. S'il rend impossible la bonne exécution de la tâche il faut impérativement gérer cet imprévu, mais s'il ne fait qu'amoindrir les performances (e.g. temps d'accomplissement), le choix de prendre en compte ou non un imprévu sera le résultat d'un compromis entre performance et complexité de la programmation.

Dans tous ces cas, l'imprévu est toujours considéré comme un effet *parasite.* La correction, si nécessaire, se fait au niveau du robot si on considère que les conditions qui l'ont générée doivent être prises en compte, ou au niveau de l'environnement si on considère que l'erreur a été provoquée par une situation anormale qu'on ne veut pas prendre la peine de savoir gérer.

Cependant, une façon de “résoudre” un problème donné est de considérer que ce problème est mal formulé. Les deux “solutions” ci-dessous ont la particularité de ne plus considérer les phénomènes imprévus comme des parasites à corriger, mais comme des objets d'étude, voire une richesse potentielle dont tirer parti :

- On constate expérimentalement que le robot n'est pas capable, par cette méthode, d'accomplir la tâche assignée (se rendre dans le hall d'entrée). Cependant, on constate qu'il réalise bien une autre tâche, qui est de se rendre à la porte. On redéfinit alors a-posteriori les buts du robot comme étant ce qu'il s'est révélé concrètement capable d'effectuer.
- On cherche comment, dans une autre situation, *tirer parti* du fait de savoir glisser sur les pas de portes. Ce peut être par exemple une méthode pour détecter les pas de portes — et donc un certain nombre de portes. En effet, là où le robot glisse systématiquement, il a de bonnes chances de se trouver à un pas de porte. On change alors éventuellement les buts du robot pour être capable de tirer parti de cette possibilité découverte empiriquement.

Cette dernière possibilité peut se présenter même hors de tout défaut réel dans le comportement du robot, il suffit que se présente une situation non prévue par le programmeur, même si elle ne remet pas en cause le programme existant. Typiquement, le cahoutchouc des roues est en général bien adhérent et ne glisse pas. Ce qui se passe alors, c'est que le pas de porte bloque une des roues tandis que l'autre le franchit, puis la première rattrape son retard. Le retour à la trajectoire planifiée se joue donc simplement au niveau des asservissements. Cependant, même si cela n'a que rarement de conséquences sur l'accomplissement du but, le mouvement du robot aux pas de portes a des particularités très apparentes pour un observateur humain. On pourrait alors imaginer de tirer parti de cette observation pour penser à faire reconnaître les pas de porte au robot. La constatation d'une propriété inattendue du mouvement, même si elle n'empêche pas le robot d'effectuer sa tâche, est à voir une manifestation de l'imprévu.

Ces solutions inhabituelles consistant à changer les fonctionnalités du robot permettent alors de prendre en compte un troisième critère :

- *Souplesse des fonctionnalités* : les fonctionnalités initialement prévues peuvent être plus ou moins modifiables. Les deux extrèmes sont d'une part des fonctionnalités impératives, aucunement remettables en cause, d'autre part des fonctionnalités simplement indicatives, mais qu'il n'est pas trop génant de redéfinir si les possibilités d'évolution que cela ouvre semblent compenser la gène que cela apporte.

Ce dernier critère, peu reconnu par rapport aux deux premiers, nous semble particulièrement important pour toute approche incrémentale. *Admettre une certaine souplesse dans les fonctionnalités du robot, c'est se donner la possibilité de considérer l'imprévu comme une richesse potentielle à prendre en compte, c'est redonner un rôle à la créativité du concepteur dans la programmation des robots. Cette démarche demande de ne plus aborder la programmation de robots autonomes exclusivement en termes de “résolution de problèmes”, mais de considérer ce que le robot sait déjà faire comme plus important que ce qu'il ne sait pas encore faire* (voir Agre 1988).

C'est cela que nous pensons pouvoir aborder par l'incrémentalité. Pour nous, ce qui distingue la notion de mise-au-point de celle d'incrémentalité, c'est que l'expérience ne sert pas qu'à corriger les erreurs, mais réellement à modifier la définition et la *structure* du robot. Les fonctionnalités peuvent changer en même temps que l'implémentation lors d'un tel développement. Un robot n'est plus

essentiellement le résultat d'une conception passée, mais est abordé dans la perspective d'un développement continu. C'est ce que nous voulons résumer en disant que le problème que permet d'aborder la démarche incrémentale est justement celui de l'imprévu.[18]

Une note sur l'incrémentalité de Steels

Nous avons présenté notre approche par contraste avec celle de Brooks, qui est la plus connue et la plus acceptée. Cependant, nous connaissons en robotique une autre approche dont nos préoccupations sont plus proches : celle de Luc Steels à Bruxelles (Steels 1993).

Les réflexes de Brooks étaient programmés en boucle ouverte. Steels propose, lui, la notion de contrôle par *régulation* (asservissement) comme base d'un système de réflexes. Le système global fonctionne donc en boucle fermée, et les événements pertinents auxquels le robot doit réagir agissent comme des *perturbations* sur les régulations, transmises par des capteurs. La “brique de base” de l'approche de Steels est une boucle de régulation, dont certains paramètres peuvent éventuellement être contrôlés de l'extérieur.

Cette démarche est très différente de celle de Brooks. La programmation se fait en combinant des structures génériques, tandis que l'architecture subsumption est totalement ad-hoc. De plus, le réflexe de base est en boucle fermée chez Steels (régulation), alors que l'architecture subsumption est plutôt conçue en boucle ouverte (réponse à des stimuli). Nous décririons en fait l'architecture subsumption comme un langage de programmation parallèle, alors que la démarche de Steels semble plus orientée par des préoccupations méthodologiques, telle que la formalisation de la notion de fonctionnalité émergente (Steels 1990) ou la possibilité d'une évolution structurelle systématique en vue d'une approche sélectionniste (Steels 1993).

F. LA REPRÉSENTATION PRÉDÉFINIE

Le but de cette section est de présenter la notion habituelle de représentation, que nous appelons “représentation prédéfinie”. Ce sujet peut sembler parachuté au milieu du chapitre de façon un peu incongrue, mais il en est en fait l'aboutissement logique.

Notre réflexion s'est terminée, dans la section précédente, sur l'envie de réintroduire la notion de représentation dans la problématique de l'incrémentalité. Il est donc utile de développer la notion classique de représentation pour plus tard pouvoir lui opposer notre propre approche. Nous faisons cela au §F.1. De plus, cela nous offre une perspective supplémentaire sur le rapport entre représentation et incrémentalité : nous considérons en effet que tous les problèmes décrits jusqu'ici, conditions de validité, modularité et imprévu, ne sont que les manifestations d'un même problème plus fondamental, qui est l'inadéquation de la représentation prédéfinie à la robotique autonome. Cette opinion est développée au §F.2, et

---

[18] Nous nous écartons en cela des motivations de Brooks. La méthodologie de ce dernier, si elle admet de rajouter des fonctionnalités au fur et à mesure, ne permet pas l'évolution a-postériori des buts que le robot sait déjà atteindre. Or, le développement du robot peut rendre les buts existants moins intéressants, et peut inciter alors à réutiliser autrement les structures acquises. Cet aspect de l'incrémentalité nous intéresse beaucoup.

constitue une justification supplémentaire de notre idée d'aborder l'incrémentalité par le biais d'une nouvelle approche de la représentation.

F.1. La notion de représentation prédéfinie

Pour présenter ce que nous appelons la représentation prédéfinie, il semble utile dans un premier temps de distinguer deux aspects dans la modélisation. En effet, si certains modèles n'ont pour but que de “coller aux données” dans une situation donnée (modèles empiriques), d'autres ont comme intérêt principal une formulation synthétique qui les rend génériques à un ensemble très étendu de situations (modèles explicatifs). La représentation prédéfinie est plus particulièrement liée à ce dernier aspect, et sera donc plus facile à décrire dans le cadre des modèles explicatifs. Cependant, elle se manisfeste très souvent aussi dans le cas des modèles empiriques.

Modèles explicatifs et modèles empiriques

Dans les modèles utilisés en géométrie algorithmiques, les paramètres utilisés sont des grandeurs physiques : des mesures de distances entre points d'un espace cartésien. Un tel modèle sera appelé “explicatif”.[19]

Plus précisément un modèle explicatif est un modèle dont les paramètres sont dits “réifiables” (Richalet 1991), c'est-à-dire dont les propriétés sont déjà bien répertoriées, et qui ont été utilisés dans d'autres modèles. Par exemple, un paramètre décrit comme une “force”, c'est-à-dire auquel on attribue toutes les propriétés connues définissant une force, est réifiable, car on lui attribue des propriétés plus générales que celles utilisées par le modèle particulier élaboré, et cela permet éventuellement de généraliser celui-ci à d'autres circonstances.

Un modèle explicatif, donc, ne sert pas seulement à prédire des valeurs numériques, il a aussi un aspect qualitatif, puisqu'on peut le mettre en rapport avec d'autres modèles (ceux qui s'expriment avec les mêmes grandeurs réifiables).

Les grandeurs physiques sont le type même de grandeurs réifiables, mais nous pouvons considérer aussi toute grandeur symbolique comme réifiable. Par exemple, modélisons la situation de la figure 4 par des prédicats du modèle STRIPS (Fikes & Nilsson 1971) :

`SUR(cube1,cube2)`, `SUR(cube2,table)`, `LIBRE(cube1)`, et `PINCE_LIBRE()`.

Ces grandeurs, quoique non physiques mais logiques, ont toutes les qualités pour être réifiables.

---

[19] Ce terme n'est pas usuel, non plus que celui de modèle empirique. Dans les sciences de l'ingénieur, on trouve plutôt les termes “modèles de connaissance” et “modèles de représentation” (Richalet 1991). Nous avons préféré les éviter dans le cadre des sciences cognitives, où la référence à des notions comme connaissance et représentation serait malvenue. En IA et en philosophie, on trouve souvent “modèles analytiques” pour les modèles explicatifs, appellation que nous avons également évitée, le terme “analytique” étant une source de confusion pour les mathématiciens et physiciens.

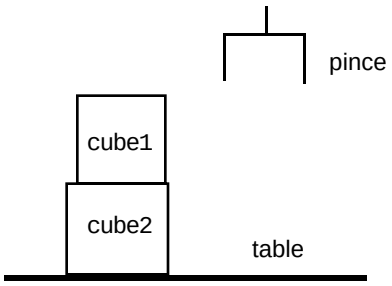


Figure 4. Un environnement représentable par des prédicats du modèle STRIPS.

Un modèle empirique, quant à lui, est une représentation mathématique plaquée sur des données expérimentales, une “boîte noire” liant certaines variables, et dont les paramètres n'ont pas d'interprétation particulière — ils n'ont qu'une valeur locale, non réifiable. Un perceptron, par exemple, est un modèle empirique, dont les poids des connexions constituent les paramètres. La régression des statisticiens est un autre exemple.

Contrairement à un modèle explicatif, un modèle empirique, ne s'intéresse donc qu'à l'aspect quantitatif d'un phénomène : il s'agit de prédire des valeurs numériques, et rien d'autre.

Quoique cette distinction entre modèles empiriques et explicatifs puisse être jugée trop caricaturale (nombre de modèles tenant des deux aspects à la fois), elle explicite bien les deux rôles principaux d'un modèle lorsqu'il s'agit de construire un système : faire des prédictions sur un phénomène, et en fournir un compte rendu synthétique (réutilisable dans d'autres contextes).

La représentation prédéfinie dans les modèles explicatifs

Pourquoi les modèles explicatifs sont-ils si naturels à utiliser ?

Un modèle explicatif est orienté vers le confort du concepteur : la facilité de spécifier des buts, de reconnaître des situations d'échec ou de succès, d'exprimer des heuristiques, d'expliquer formellement ce que fait un algorithme, de prouver sa convergence, et surtout de décomposer hiérarchiquement un problème en sous-problèmes (analyse descendante). On représente un problème sous une forme telle qu'on sache a-priori le résoudre algorithmiquement sous cette forme.

L'idée est de *faire en sorte que la difficulté conceptuelle (par opposition à technique ou algorithmique) se concentre dans l'étape de modélisation elle-même.* Une fois un problème modélisé, sa résolution devient purement formelle, c'est-à-dire qu'elle ne concerne que les termes du modèle, indépendemment de toute référence à la situation réelle. Ceci est possible parce qu'on connaît d'avance la manipulation de ces termes et les relations qu'ils ont entre eux (termes réifiables). *En fait, un modèle explicatif se justifie et se caractérise par la connaissance préalable de la façon dont il peut être utilisé.* On dit que les termes du modèle sont “prédéfinis”.

Dans cette approche, la modélisation est difficile, l'interprétation (utilisation) est relativement facile, et les phénomènes imprévus sont vus comme des parasites.

Nous appelons “représentation prédéfinie” la notion de représentation fondée sur l'emploi de termes prédéfinis.

L'objectivisme

L'emploi de termes prédéfinis n'est pas un problème dans la plupart des domaines scientifiques car, comme nous l'avons déjà dit, quand un modèle abstrait est appliqué à une situation concrète c'est toujours par l'intermédiaire d'un humain, qui dispose de toutes ses connaissances de sens commun et de toute sa culture scientifique pour correctement interpréter et utiliser le modèle. En robotique autonome, par contre, vouloir à la fois :

[utiliser des termes prédéfinis
[et ignorer le rôle actif du concepteur dans leur application concrète

implique une position philosophique qu'en suivant Varela (1993) et Stewart (1994b) nous appellerons “objectiviste”. L'objectivisme postule qu'on peut juger de la validité d'un modèle de la nature indépendemment du concepteur qui a formulé ce modèle. Par conséquent, à un niveau de description donné, on a le droit de confondre le monde avec la connaissance qu'on en a. L'utilisation de termes prédéfinis est alors justifiée parce que ces termes sont en quelque sorte *préexistants* à l'établissement du modèle par le concepteur.

Cette position se retrouve dans d'autres domaines scientifiques, par exemple dans l'idée que les lois de la physique puissent avoir une existence objective, et que la science ne ferait que découvrir ces lois préexistantes. Néanmoins, en physique, ce débat existe depuis longtemps (Poincaré 1902), tandis qu'en robotique il reste marginal.

La représentation prédéfinie dans les modèles empiriques

La prédéfinition des termes est le fondement explicite et reconnu de la modélisation classique (explicative). Cependant, à notre avis, l'aspect prédéfini existe encore dans de nombreuses approches qui se veulent une remise en question de cette dernière.

Prenons l'exemple des réseaux neuronaux faisant de l'approximation de fonctions. Le réseau a une entrée x et une sortie y, et on cherche une relation $y=f(x)$. La fonction f elle-même n'a rien de prédéfini : sa forme paramétrique est générique, et ne reflète pas une interprétation fixe. En revanche, x et y restent des termes prédéfinis ; par exemple s'il s'agit d'apprendre une “carte” associant à une position cartésienne de l'environnement une direction privilégiée, via un apprentissage par renforcement (Sutton 1990) ou un perceptron, x reste prédéfini comme la position cartésienne du robot. Le jour où ces méthodes devront être implémentées sur un robot réel, celui-ci devra à partir de ses propres données sensorielles déterminer sa position cartésienne dans son environnement — et pas autre chose.

En conclusion, pour des modèles empiriques, les termes eux-mêmes entre lesquels on cherche la relation empirique restent prédéfinis. *L'apport des modèles empiriques sur les modèles explicatifs est pour nous de nature non pas conceptuelle mais essentiellement technique,* c'est-à-dire que la différence réside non pas dans la façon dont ils résolvent les problèmes de fond (conditions de validité, modularité, imprévu), mais dans la faisabilité de leurs mises en oeuvres respectives :

- Dans un modèle explicatif, la sensibilité du comportement du robot aux erreurs d'identification est grande ; en revanche les modèles empiriques ont souvent des propriétés de *robustesse* très intéressantes.

- Les modèles empiriques ont un aspect *générique* qui les rend utilisables dans nombre de situations, qui nécessiteraient autant de modélisations explicatives spécifiques.

F.2. Une “cause commune” à tous les problèmes ?

Nous avons dans les sections précédentes décrit séparément les problèmes des conditions de validité, de la modularité, et de l'imprévu. Nous montrons ici que ces problèmes sont étroitement liés : il s'agit de trois conséquences particulières, trois aspects, d'un même problème de fond, celui de la représentation prédéfinie.

Représentation prédéfinie et conditions de validité

Analyse descendante

Reprenons l'exemple du suivi de trajectoire. Nous avions précédemment proposé un programme réflexe résolvant le problème posé :

```
[3]  { Δroues (+1.25,+1.25); Δroues (-0.8,+0.8);
       Δroues (+0.5,+0.5); Δroues (+0.8,-0.8);
       Δroues (+2.5,+2.5) }.
```

Mais nous avions ensuite signalé que bien que le seul programme opérationnel soit la forme motrice [3], ce programme est souvent écrit et présenté sous une forme telle que [2’]

```
[2’] { translation (20cm); rotation (60°);
       translation (8cm); rotation (-60°);
       translation (40cm) }.
```

oubliant ainsi qu'une instruction comme `translation (20cm)` ne signifie pas que le robot fera une translation de 20 cm, mais effectuera aveuglément la commande `Δroues (+1.25,+1.25)`.

On voit bien sur cet exemple le rapport entre le problème des conditions de validité et celui des représentations prédéfinies. La notion géométrique (abstraite) de translation est celle qui nous a permis de modéliser et de résoudre le problème en premier lieu. Dans le programme résultant, nous accordons à `translation` la même signification, car c'est elle qui rend pertinente l'utilisation du modèle géométrique. Or, en pratique, `translation` *n'est plus un terme abstrait mais désigne en fait un mécanisme,* une routine informatique calculant des ordres moteurs, mais que nous aimerions, nous, programmeurs, *interpréter* comme effectuant une translation.

Si l'on fait part à cette notion d'interprétation, on voit bien que le problème des conditions de validités provient d'une interprétation inadéquate, et doit être corrigé par le concepteur en conséquence. Ne pas prendre en compte cette notion d'interprétation du concepteur, et refuser l'intervention de celui-ci pour corriger les erreurs, c'est considérer l'interprétation de `translation` en tant que “translation” comme prédéfinie et systématique — et la question des conditions de validité de cette interprétation se pose alors. Ce problème peut être illustré par la figure 5.

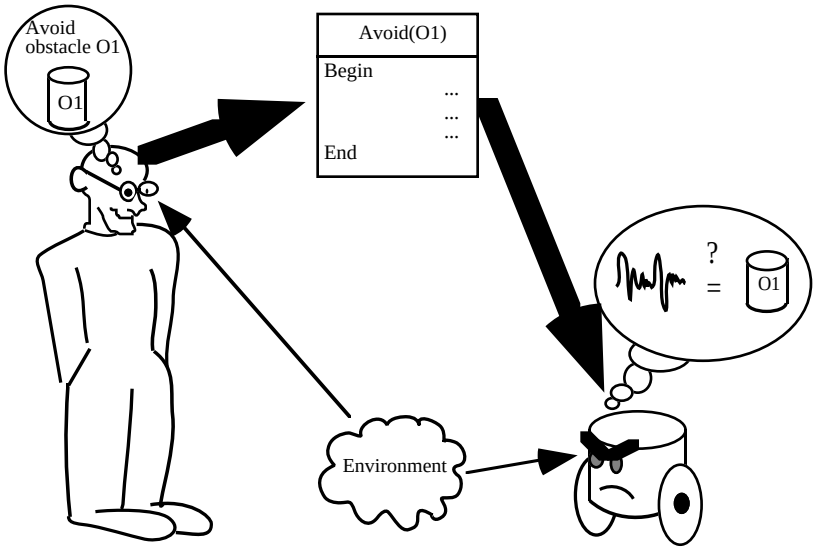


Figure 5. La représentation prédéfinie et le problème des conditions de validité. Le concepteur formule son problème en certains termes connus, tels que “obstacle“ ou “translation”, et veut que l'interprétation des termes que manipule le robot, par exemple `obstacle` ou `translation`, corresponde à leur signification prédéfinie. Il y a là une correspondance difficile à établir.

La position objectiviste résoud cette question en postulant qu'il existe une modélisation dont l'interprétation puisse effectivement être indépendante du contexte et du concepteur. Selon ce point de vue, en poursuivant la recherche et la sophistication des techniques, on convergerait à terme vers cette modélisation idéale, et donc le problème serait essentiellement théorique et technique.

Analyse comportementale

Revenons à présent au programme résultant de l'analyse comportementale :

[4] Δroues(f(cellules) $\oplus$ g(proximètres))

Ce programme, s'il est bien implémenté directement en termes moteurs Δroues, n'a certainement pas été conçu en ces termes. Pour déterminer en pratique f et g, le concepteur va plutôt raisonner ainsi : la vitesse de translation V du robot est choisie constante, suffisament faible pour que l'évitement d'obstacle ou l'attirance vers le but puissent ne se programmer que sur la rotation Ω. Cette rotation ira, pour le réflexe f, dans la direction du but, et pour le réflexe g dans une direction opposée à l'obstacle.

Le programme sera alors plus souvent exprimé sous la forme [6] suivante que sous la forme [4] :

[6] $\left[\begin{array}{l} V \leftarrow \text{constante tant que but non atteint} \\ \Omega \leftarrow \text{fonction des cellules; } \Omega \leftarrow \text{fonction des proximètres} \end{array}\right.$

cette notation indiquant comment V et Ω sont calculées à chaque instant (la composition des réflexes utilisant des ressources partagées étant implicite).

Le programme [6] reste un programme moteur si V et Ω sont définis comme suit : pour une commande Δroues(x,y), V désigne x+y et Ω désigne x-y. Il s'agit juste d'un changement de variables formel ; V et Ω ne désignent *pas* les vitesses de translation et rotation du robot dans son environnement, mais de vraies commandes motrices. Pour concevoir ce programme, nous, concepteurs, avons *interprété* V et Ω comme des vitesses de translation et rotation, adoptant certaines hypothèses

implicites pour faciliter notre raisonnement — mais cet intermédiaire de conception est absent du programme final.

Cependant, le programme [6] poserait les mêmes problèmes que le programme [2'] à partir du moment ou V et Ω seraient censées désigner *effectivement* les vitesses de translation et de rotation du robot, de même que dans le programme [2'], les termes `rotation` et `translation` représentaient les concepts géométriques de translation et rotation, c'est-à-dire un niveau de description abstrait.

Dans la pratique, c'est souvent ce qui se passe. On a tendance à parler de comportements définis par des fonctionnalités prédéfinies (aller à un but, éviter les obstacles, fuir des prédateurs, explorer l'environnement, se reposer) que de comportements définis intrinsèquement par leurs mécanismes (réflexes). La notion de comportement est alors une notion abstraite ; elle est à l'analyse comportementale ce que les niveaux de description sont à l'analyse descendante.

Réprésentation prédéfinie et modularité

Nous avons décrit le problème de la modularité comme la difficulté de réellement séparer l'implémentation de deux niveaux de descriptions (ou comportements). Ce problème concerne donc le rapport de deux abstractions entre elles, tandis que le problème des conditions de validité concernait le rapport entre une abstraction et le monde réel.

L'emploi de termes prédéfinis fixe la “signification” d'un niveau de description, dans le but de rendre la manipulation de ses termes indépendante de toute autre considération. Lesdites “autres considérations” peuvent être la nécessité de faire référence à un monde concret, et cela pose le problème des conditions de validité, ou bien la nécessité de collaborer de façon cohérente avec d'autres niveaux de description prédéfinis, ce qui pose le problème de la modularité.

Il y a un problème dans les deux cas, et nous prétendons qu'il est de même nature, c'est-à-dire que le problème de la modularité est comme celui des conditions de validité une conséquence de l'emploi de termes prédéfinis. Par exemple, une fois que l'on a affirmé que la capacité de navigation d'un robot en environnement encombré peut se définir complètement dans des termes géométriques, peu importe pour utiliser ce modèle que les volumes soient donnés a-priori, ce qui pose le problème des conditions de validité, ou analysés à un niveau inférieur de façon à les déterminer autrement (e.g. géométrie projective si l'on pense à une caméra), ce qui pose le problème de la modularité entre les deux niveaux. En définitive, la validité d'un niveau de description prédéfini ne dépend pas du fait qu'il soit le dernier ou non de la hiérarchie des niveaux, et ce soit ou non lui qui soit censé ou non devoir “refléter fidèlement le monde réel”. Le problème de la modularité est au fond le même que celui des conditions de validité.

Représentation prédéfinie et imprévu

Le problème de l'imprévu est le suivant : nous avons modélisé un problème en termes prédéfinis, et un phénomène imprévu se produit alors qu'il faut prendre en compte. Comment alors le prendre en compte ?

Modéliser un problème en termes prédéfinis, c'est faire un choix des termes exclusifs qui caractériseront ce problème, et à partir de là, même si nous sommes conscients des imperfections du modèle, *nous ne pouvons l'utiliser par un processus*

automatique qu'en faisant l'hypothèse implicite qu'il est complet. Si alors les conditions de validités ne sont plus vérifiées, les prédictions quantitatives seront automatiquement considérées comme invalides. Ce cadre ne permet pas de gérer les éventuelles solutions opportunistes à l'imprévu, du type “considérer que le robot ne fait pas ce que l'on veut mais fait néanmoins quelque chose d'intéressant que nous allons exploiter”, car cela impliquerait de considérer les prédictions comme valides, mais au prix d'une autre interprétation pour certains termes du modèle, qui alors ne seraient plus prédéfinis.

G. CONCLUSION : NOTRE PROBLÉMATIQUE

Dans ce chapitre 2 nous avons développé une réflexion que nous devons à présent cristalliser en une problématique particulière. Cette réflexion comportait trois volets :

- Un volet décrivant le principe des approches habituelles de la robotique (sections A, B, D),
- Un volet critiquant la pertinence de ces approches pour aborder la robotique *autonome* (sections C, D, F),
- Un volet proposant des pistes pour aborder autrement celle-ci (section E).

C'est le troisième volet que nous devons synthétiser ici en une problématique. Les notions mises en avant étaient celles d'incertitude (§E.1), d'incrémentalité et de représentation (§E.2). Un premier niveau de problématique est alors le suivant :

> *Problématique générale* : Nous voulons introduire le cycle de l'incrémentalité entre conception et adaptation, en tant que façon de gérer l'imprévu. Nous voulons de plus aborder cette incrémentalité via la notion de représentation, car elle est au coeur du problème et que l'expliciter est le meilleur moyen de ne pas la refuser ou l'éluder. Enfin, nous voulons explorer en quoi la gestion explicite de l'imprévu peut être une façon de gérer le problème des conditions de validité.

On retombe ainsi sur la formulation du chapitre 1.

Cette problématique révèle la volonté de “mettre les pieds dans le plat” pour aborder les difficultés soulevées tout au long de ce chapitre (avec un bémol pour le problème de la modularité, qui n'est pas explicitement abordé au contraire de ceux des conditions de validité ou de l'imprévu). Elle reste cependant encore trop générale et trop riche, et plus concrètement nous n'allons dans la suite de cette thèse développer que quelques points :

Concernant la gestion de l'incertitude

- Pour automatiser la gestion de l'incertitude, il faut un cadre formel.
- Il faut alors montrer en quoi sa mise en oeuvre répond au problème des conditions de validité.

Le chapitre 3 propose une théorie de l'incertitude, et le chapitre 4 montre une première mise en oeuvre sur des expériences simples de robotique.

Concernant l'incrémentalité

- Pour rejeter la représentation en termes prédéfinis sans pour autant rejeter la notion même de représentation, il faut proposer pour celle-ci une approche alternative.
- Il faut alors proposer un cycle incrémental fondé sur cette approche et qui réponde au problème de l'imprévu.

Nous procédons en fait inversement : nous décrivons d'abord une méthodologie pour l'incrémentalité, illustrée par des expériences (chapitre 5), et à partir de là nous proposons la notion de “représentation contingente” (chapitre 6).

H. COMPLÉMENTS

Il nous reste, pour boucler la réflexion de ce chapitre, trois remarques importantes bien que non nécessaires pour définir notre problématique.

Premièrement, nous voulons mettre en gardre contre certaines métaphores biologiques. Les êtres vivants sont une source d'inspiration importante pour la robotique autonome, et à raison puisque ce sont leurs performances que l'on cherche en fin de compte à reproduire. Il ne faut cependant pas pousser trop loin l'analogie, comme nous l'expliquons au §H.1.

Deuxièmement, nous voulons dire un mot de la *simulation* comme outil d'étude de la robotique autonome. Souvent, l'emploi d'une simulation, loin de constituer une étude préliminaire simplifiée d'un problème de robotique (comme elle le devrait), revient à changer radicalement le problème étudié. Nous voulons citer alors quelques précautions à prendre pour éviter cela. C'est l'objet du §H.2.

Troisièmement, les termes que nous avons choisi d'utiliser ne sont pas très habituels. Une grande partie des thèmes de ce chapitre ont déjà été abordés dans la littérature, mais en parlant de symboles plutôt que de termes prédéfinis, de problème de l'ancrage plutôt que de conditions de validité, etc… Nous décrivons au §H.3 la façon classique de présenter ces préoccupations, et les raisons pour lesquelles nous avons choisi de ne pas faire référence aux termes consacrés.

H.1. Le danger des métaphores biologiques

En biologie, on distingue la phylogénèse (l'évolution des espèces) de l'ontogénèse (le développement des individus). Étant donné un individu particulier, on appelle “génome” l'ensemble de ses gènes, et on appelle “phénotype” ses caractéristiques biologiques (par exemple anatomiques, physiologiques, comportementales…). Le génome est hérité de son ou ses parents par un processus de phylogénèse, et le phénotype est élaboré au cours d'un processus d'ontogénèse mettant en jeu le génome et l'environnement.

Phylogénèse et ontogénèse sont vus en général comme deux problèmes distincts, faisant appel à des mécanismes indépendants, selon la position suivante :

- La phylogénèse d'un organisme est entièrement résumée par son génome ; c'est-à-dire que quel que soit le mécanisme ayant permis d'obtenir ce génome, on peut en faire abstraction pour décrire le rôle des gènes dans l'élaboration de l'anatomie, de la physiologie et du comportement de l'organisme.
- Réciproquement, l'ontogénèse n'a aucune influence sur le mécanisme qui fournit les génomes. L'influence de l'ontogénèse sur la phylogénèse ne se fait pas au niveau d'un mécanisme génétique, mais indirectement par la sélection naturelle.

La phylogénèse et l'ontogénèse se situent sur des échelles de temps distinctes. L'ontogénèse se fait sur la durée de vie d'un individu, tandis que la phylogénèse ne se fait sentir que sur de nombreuses générations.[20]

Un robot n'est pas le produit d'une phylogénèse. C'est un concepteur humain qui a déterminé sa forme physique, ses capteurs, ses actionneurs, ses prétraitements sensoriels, son architecture logicielle, ses mécanismes adaptatifs éventuels. Il est néanmoins tentant de voir une analogie entre d'une part la conception d'un robot et la phylogénèse, et d'autre part l'adaptation d'un robot et l'ontogénèse.

Cette analogie associe le génome avec la structure, et le phénotype avec les valeurs particulières des paramètres. Cela reflète l'idée traditionnelle que le génome “code” la morphologie d'un organisme, ses facultés sensorielles et motrices, ainsi que certains comportements et mécanismes adaptatifs “innés”. Dans un robot, tous ces aspects sont justement ceux que doit spécifier le concepteur — c'est ce qui justifie l'analogie. L'intelligence et la créativité humaines remplaceraient en somme des millénaires d'évolution “aveugle” pour concevoir des robots de façon adéquate.

Si l'on fait confiance à cette analogie, il semble qu'on puisse en toute confiance séparer les problèmes de la conception et de l'adaptation et les aborder indépendemment, *puisqu'après tout cette stratégie a fait ses preuves chez les êtres vivants.* C'est là, pensons-nous, le préjugé implicite qui explique que l'idée de séparer les deux problèmes (conception et adaptation) soit rarement remise en question.

Or, même en biologie, il n'est pas prouvé que le génome d'un organisme “code” sa structure — i.e. ce qui est invariant — et que l'individu (phénotype) résulte de l'adaptation de paramètres plastiques déterminés par cette structure. La séparation, entre éléments “innés” et éléments “acquis” est fortement questionnable.

D'abord, le rôle du génome ne se manifeste dans l'organisme que par des phénomènes extrèmement complexes (cf McFarland 1987 à “Ontogénèse”). L'ontogénèse est bien plus qu'un simple processus d'adaptation aux circonstances locales, c'est elle qui détermine l'existence même de l'organisme au bout du compte (cf Stewart 1993a). Les notions d'inné et d'acquis seraient des distinctions très simplificatrices, dans lesquelles il ne faudrait voir aucune réalité fondamentale (Stewart 1993b). Dans la pratique ces notions sont de toute façon délicates à manipuler : comme le fait remarquer Stewart (1993b), la théorie génétique ne peut que nous renseigner sur la dépendance de la *variation* de certains caractères avec la *variation* de certains gènes, et cela ne suffit pas à dire que ces caractères aient été programmés dans les gènes, ni, surtout, que les caractéristiques *invariantes* d'une

---

[20] Notons qu'entre ces deux échelles de temps on peut vouloir placer une évolution des habitudes sociales (Mc Farland 1987, à “comportement acquis”). Par la suite nous oublierons cette dimension sociale, car la robotique visée par nos métaphores biologiques est essentiellement celle des robots individuels et non celle des collectivités de robots.

espèce soient déterminés par les gènes. L'idée d'un “programme génétique” pose à cet égard de nombreux problèmes.

Si la distinction de fond entre phylogénèse et ontogénèse n'est pas remise en question, les métaphores qui y font référence (surtout dans le domaine de l'informatique) semblent de plus en plus discutables, car elles n'en retirent souvent que les aspects caricaturaux — telles les notions d'inné et d'acquis. Pour notre part, nous arrêterons là nos métaphores biologiques.[21] Le cycle de l'incrémentalité pourrait, dans une certaine mesure, être mis en parallèle avec les phénomènes de maturation (souvent oubliés lorsque l'on schématise l'ontogénèse comme faisant de l'adaptation). Mais nous n'avons pas besoin de ces analogies dangereuses. *La problématique de la conception de robots peut être décrite en soi, c'est un problème à part entière, et non pour nous une façon détournée d'aborder par le travers les problèmes du vivant.*[22]

H.2. Le danger des simulations

En robotique autonome, une simulation vise à remplacer l'environnement du robot, afin d'éviter la gestion longue, difficile et coûteuse d'un robot physique. Un système simulé est alors composé de trois modèles :

- Un modèle simulant l'environnement indépendemment du robot (par sa dynamique interne),
- Un modèle simulant les interactions du robot avec l'environnement,
- Le modèle de fonctionnement du robot lui-même.[23]

Une simulation est un environnement de nature *formelle.* Toute nos réticences sur les représentations prédéfinies s'effondrent : le modèle du monde *est* préexistant au robot, et on peut légitimement adopter un point de vue objectiviste. Les hypothèses implicites attachées à un modèle de connaissance peuvent être explicitées.

L'utilisation d'une simulation permet ainsi très naturellement d'éluder les problèmes principaux de la robotique autonome, puisque l'environnement du robot n'est pas distinct de la connaissance que peut en avoir son concepteur. Pour faire des simulations un outil pour l'étude de la robotique autonome, cette distinction doit

---

[21] Notons que les métaphores biologiques sévissent également au sujet de l'incrémentalité. Brooks (1991) par exemple considère que l'évolution des êtres vivants consiste en des cycles de modification et de mise-au-point, et affirme par conséquent : “we are not so concerned that it might be dangerous to test simplified Creatures first and later add more sophisticated layers of control because evolution has been successful using this approach”. Cette analogie nous paraît tout aussi douteuse que celle que nous avons dénoncée plus haut.

[22] Cela ne signifie pas que le roboticien ne puisse pas s'inspirer de la biologie. Nombre de techniques d'apprentissage sont nées d'inspirations biologiques (et notamment neurophysiologiques). Lorsqu'il s'agit de réaliser un robot autonome, ces techniques sont toutefois plus utilisées du point de vue de l'ingénieur (construire un système qui réalise certaines tâches) que du biologiste. Par exemple, beaucoup de biologistes contestent la pertinence des réseaux neuronaux ou des algorithmes génétiques en tant que modèles des phénomènes biologiques, mais pour l'ingénieur il s'agit de techniques d'optimisation efficaces et donc intéressantes. La réalisation de systèmes robotiques mettant en oeuvre ces techniques peut également être une façon de valider des hypothèses biologiques (Webb 1995), mais il s'agit alors d'une toute autre problématique que la nôtre.

[23] En robotique, ce modèle ne doit pas être considéré comme une simulation, puisque la même structure serait utilisée dans une application réelle. Ce serait différent dans un cadre éthologique, où l'organisme artificiel est censé représenter un organisme vivant. Le modèle de fonctionnement de l'organisme deviendrait alors un troisième niveau de simulation.

être rétablie par certaines précautions méthodologiques, que nous détaillons à présent en les opposant aux pratiques habituelles.

Le réalisme est une mauvaise politique

Reprenons notre problème de navigation, le robot pouvant par ailleurs être muni des capteurs que l'on jugera nécessaires. Imaginons que pour simplifier l'étude préliminaire du problème on cherche à simuler cet environnement, et qu'il s'agisse plus précisément d'étudier ainsi l'emploi d'un modèle polyédrique.

La simulation sera alors dans l'extrême majorité des cas elle-même fondée sur un modèle polyédrique. L'environnement est modélisé en fonction du modèle même dont il s'agit d'étudier les performances… Le problème est alors que les paramètres du modèle interne et de l'environnement (formel) restent qualitativement en correspondance un à un. Cela revient à supposer l'existence d'un “capteur de face d'obstacle” ou d'un “capteur d'approximation polyédrique d'objet”, dont on ne dispose évidemment pas.

On pourrait supposer qu'un moyen de tenir compte de cela serait d'introduire dans le modèle interne du robot un certain bruit par rapport au modèle simulant l'environnement, qui représenterait alors “l'incertitude” de la modélisation effectuée. *Cependant, les paramètres du modèle et de l'environnement restent en correspondance qualitative,* et le fait que leurs valeurs numérique puissent ne pas parfaitement coïncider ne change guère les choses : on ne dispose toujours pas réellement de “capteurs d'approximation polyédrique d'objet”, même bruités. Le problème des conditions de validité est qualitatif avant d'être quantitatif.

On peut pour remédier à cela simuler l'utilisation de capteurs “réalistes” tels que des proximètres. Le robot ne connaîtra de son environnement que la distance des obstacles selon certaines directions fixées.

Or là encore le résultat est biaisé. Dans la simulation on calculera ces valeurs de capteurs en fonction directe (éventuellement bruitée) des distances dans le modèle géométrique. Les proximètres sont simulés comme de *vrais* “capteurs de distance” et non des capteurs infrarouges. C'est encore l'environnement qui est conçu en fonction du modèle interne du robot, et c'est là que se situe le problème de fond. Le robot utilisait quelque part un algorithme conçu par référence à une notion de distance. Si on mesure la distance directement sur un modèle formel, le problème des représentations prédéfinies est éludé. Si on tente encore de pousser le réalisme, on finit par douter de l'intérêt principal de la simulation, qui est censé être sa simplicité. Par exemple, on peut toujours modéliser l'émission d'infrarouges et sa réflection sur les faces des obstacles, en introduisant des incertitudes dans les calculs à divers niveaux : direction et puissance initiale de l'émission, étalement spatial de la réflection, position et fiabilité du récepteur. Mais à ce niveau de détail la complexité de la simulation devient très grande, alors que d'autre part rien ne garantit malgré tout que la solution adéquate pour l'environnement simulé pourra, aux ennuis techniques près, l'être pour un environnement réel. En effet certains paramètres essentiels ont encore été oubliés : couleurs, textures (un environnement monochrome et monotexture n'est guère naturel), voire température des obstacles et du sol, interférence entre les émissions infrarouges, incertitudes sur la position même du robot, disparité entre les capteurs faussant la calibration, etc…

Le réalisme semble en définitive une mauvaise stratégie à adopter. À moins de programmer une simulation aussi complexe que l'environnement dont on cherchait à

simplifier l'étude, il semble présomptueux de considérer une simulation comme une version simplifiée de la situation réelle correspondante. Un problème posé en simulation *est avant tout un problème formel.* Si l'on nomme les éléments de ce problème “distance”, “profondeur stéréoscopique”, “obstacle”, “motivation”, “faim”, “nourriture”, “prédateur” ou “femelle”, c'est avant tout pour que les autres personnes comprennent, par analogie, la structure du problème. Cela ne suffit pas à assurer qu'un robot fonctionnant dans la simulation aura un comportement analogue dans la réalité.

Cette position est, bien entendu, liée à notre acception particulière de la robotique autonome et à nos préoccupations. Si la simulation est utilisée comme un outil préalable à la construction d'un robot réel, et non destinée à prouver en elle-même quelque chose de réaliste, alors il s'agit d'une méthodologie très appréciable, voir par exemple le travail de Pichon (1991) en tant que fondement du robot développé par Blanes (1991).

Quelles simulations intéressent la robotique autonome ?

L'intérêt d'une simulation n'est guère dans son “réalisme” fonctionnel plus ou moins poussé, mais dans les types de problèmes que l'on peut s'y poser et résoudre ainsi. Ces problèmes peuvent être très instructifs pour un concepteur ; il peut tirer de leur étude une certaine expérience abstraite, qu'il peut ensuite adapter (pas automatiquement mais en intervenant en tant qu'humain) pour résoudre certains problèmes concrets. Un environnement simulé serait alors à voir comme une “écologie artificielle”, une façon d'étudier dans un cas simple certains problèmes précis, et non de tester une forme de réalisme. Quelques rares travaux en simulation “robotique” restent pertinents de ce point de vue (Cliff 1990, Cliff & al. 1992a, Pierce 1991).

Quelles précautions doit-on alors prendre pour utiliser les simulations en robotique autonome ? La condition de base est que le modèle de l'environnement soit réellement conçu indépendemment du fonctionnement du robot lui-même. Pour cela il faut tenir compte de deux aspects :

- D'une part, si les termes que le robot manipule sont présents dans l'environnement simulé, une grande partie du problème est résolu. Un robot utilisant un modèle géométrique ne devrait pas être testé dans un environnement lui-même simulé en termes géométriques. Il faut que les mécanismes sous-jacents soient à un niveau de complexité supérieur à la notion formelle de distance, et que la valeur accessible au robot ne soit qu'une interface.
- D'autre part, si par exemple les phénomènes de glissement ne sont pas modélisés dans une simulation sous le prétexte qu'ils sont censés ne pas advenir — constituer une situation d'exception —, leur influence est totalement gommée. Il faut, au contraire, que la simulation soit *plus riche* que la formalisation que le robot utilise, pour permettre de déjà étudier, sur la simulation, la généricité de cette formalisation et sa robustesse à certains aléas.

En conclusion, *la complexité d'une simulation ne devrait pas résulter d'un souci de détail dans le réalisme, mais d'un souci de richesse relative des interactions qui définissent l'environnement.* Mathématiquement, ces dernières peuvent être simples, et d'ailleurs les simulations qui selon nous apportent quelque chose à la robotique

autonome ne sont pas les plus complexes mais sont souvent très simples (e.g. Pierce 1991).

H.3. Note historique et terminologique

En IA ou en sciences cognitives, la terminologie est une source inépuisable de malentendus et d'incompréhension. Nous avons constaté à maintes reprises que l'emploi de termes trop connus (i.e. “ancrage des symboles”) avait tendance à évoquer dans l'esprit du lecteur des idées qui occultaient notre optique un peu particulière, aussi avons nous préféré, dans ce document, inventer nos propres termes plutôt que risquer d'employer des termes consacrés.

Ce paragraphe explique comment les thèmes abordés dans ce chapitre sont plus fréquemment présentés.

L'approche cognitiviste

Les questions que nous avons évoquées ne sont pas neuves, et ont été abordées par de nombreux auteurs (citons Dreyfus 1979, Winograd & Flores 1987, Varela 1988, Stewart 1994, Reeke & Edelman 1988, Malcolm & Smithers 1990, Harnad 1990). Cependant, la cible est traditionnellement restreinte à l'approche dite “cognitiviste”. Le cognitivisme est un cas particulier d'approche objectiviste, qui a dominé l'IA pendant longtemps, et qui reste une référence incontournable. Cette importance historique nécessite un développement même succint, qui nous permettra ensuite de montrer en quoi notre volonté d'élargir le débat à d'autres approches nous a obligés à prendre certaines précautions.

Le cognitivisme : les symboles comme fondement de la cognition

Le cognitivisme[24] est une position explicitement objectiviste, qui postule que la modélisation d'un environnement en termes prédéfinis est le fondement même de la cognition (Fodor & Pylyshyn 1988). Les termes formels auraient une réalité physique à l'intérieur du cerveau, c'est-à-dire que par exemple une description symbolique ou une carte géométrique de l'environnement pourraient être mises en correspondance avec des structures cérébrales. Cela implique l'existence de modèles universels, génériques à de nombreux environnements comme à de nombreuses fonctionnalités dans ces environnements.

Le cognitivisme propose plus précisément les symboles formels comme termes de base de la cognition. Les symboles formels sont les objets manipulés par la théorie des systèmes formels. L'ordinateur serait de ce point de vue la métaphore idoine pour décrire le fonctionnement du cerveau.[25] Les symboles auraient une existence

---

[24] On trouve parfois “computationnalisme”, “symbolisme”, ou “fonctionnalisme”. Le computationnalisme met en avant le postulat que l'ordinateur est la métaphore adéquate pour le décrire le fonctionnement du cerveau ; le symbolisme s'attache particulièrement à l'utilisation de symboles formels comme base de la cognition ; et le fonctionnalisme est pour nous mal nommé puisque l'aspect fonctionnel de la cognition se trouve même dans des travaux non cognitivistes. Nous préférons pour notre part “cognitivisme”, qui dénote la volonté de poser un principe sur la nature de la cognition.

[25] Cette métaphore a largement débordé du cadre des sciences cognitives pour être adoptée dans la vie courante. Un exemple typique : dans le numéro de Libération du 27 septembre 1994, on peut lire à propos des insectes sociaux (fourmis et abeilles) : «Après les avoir longtemps caricaturés en petits robots, on les verrait plutôt maintenant comme de vrais ordinateurs sur pattes…». Cette distinction fait sourire mais fait bien

matérielle tout en restant caractérisables indépendemment du substrat biologique, de même qu'en informatique un même logiciel, même s'il doit être au bout du compte être implémenté d'une certaine façon, peut se définir indépendemment des différentes architectures matérielles permettant cette implémentation.

Ce cognitivisme pur et dur[26] a été l'hypothèse très largement dominante en IA jusque dans les années 80 (cf Dreyfus 1979). Nous le considérons ici comme une approche objectiviste radicale.

Le problème de l'ancrage des symboles

C'est dans le cadre cognitiviste qu'a été décrit ce qu'on appelle le “problème de l'ancrage des symboles” (symbol grounding problem, Harnad 1990). Le problème est qu'étant donné un système symbolique dans un environnement concret (e.g. un robot), le fonctionnement de ce système dépendra de la référence effective des symboles utilisés avec le monde réel. Il s'agit alors de trouver comment garantir cette référence automatiquement, sans faire appel à un observateur capable d'interpréter les symboles ; en d'autres termes la signification d'un symbole doit lui être *intrinsèque.*

Par exemple le sens d'un symbole formel comme `table` ou `nourriture` est par hypothèse indépendant de la façon dont il est acquis. En robotique, le problème est qu'un robot va dans la pratique avoir besoin de reconnaître une table ou de la nourriture à partir d'informations d'un niveau de description bien plus primitif. L'ancrage de ces symboles consiste à garantir que leur “sens”, tel qu'*assumé* au niveau du système formel, sera effectivement celui rendu par le mécanisme *effectif* d'acquisition de ce symbole.

Décrit ainsi, ce problème est très proche du problème des conditions de validité que nous avons décrit. Voyons à présent pourquoi nous avons évité d'y faire référence jusqu'ici.

Notre position

L'analyse que nous avons faite va bien au delà du cadre cognitiviste, pour s'appliquer, selon nous, à toutes les approches descendantes, aux approches comportementales, et à la plupart des approches adaptatives.

Cette position est en un sens très conceptuelle, mais est pourtant née du choix très pragmatique de la robotique comme cadre de réflexion. Dans cette optique, à la base, le véritable problème est que nous ne savons pas construire de robots autonome. Si nous le savions, les discussions sur l'ancrage ou les autres problèmes seraient vides de sens. Nous préférons donc aborder directement le problème en constatant qu'en pratique les robots ne font pas ce qu'on veut, et parler de conditions de validité plutôt que d'ancrage. Le message passe mieux ainsi ; avant de prendre

---

comprendre que pour beaucoup de gens, un robot n'est qu'une machine qui exécute de façon stéréotypée des tâches répétitives, tandis qu'un ordinateur a potentiellement un début d'intelligence puisque c'est un cerveau simplifié. En définitive, nous reformulerions plutôt cette phrase comme «après les avoir longtemps caricaturés en petits robots, on les verrait plutôt maintenant comme de vrais êtres vivants…».

[26] Cette position est en effet souvent affaiblie : le cognitivisme fort considère la formalisation du niveau “cognitif” comme indépendante du problème à résoudre, tandis que l'on admet de plus en plus qu'une “représentation du monde” puisse ne devoir être définie que par rapport à une certaine tâche ou fonction que doit remplir le système cognitif.

cette position terminologique nous nous heurtions au fait que l'ancrage a un sens “cognitif” bien consacré, et personne ne voulait nous accorder notre interprétation de ce problème comme équivalent à la simple constatation qu'en pratique nous ne savons pas programmer les robots autonomes en termes prédéfinis.

L'approche constructiviste

La position alternative au cognitivisme est souvent dite “constructiviste”. Nous définirions le constructivisme comme le rejet pour les sciences cognitives des approches top-down pures, de la caractérisation d'un système cognitif comme un système de traitement de l'information faisant de la résolution de problèmes, et en particulier de l'hypothèse cognitiviste proposant la métaphore de l'ordinateur pour décrire l'esprit humain.

Nous nous sentons généralement en accord avec critiques formulées par les constructivistes (Varela 1988, Varela 1989, Dreyfus 1979, Winograd & Flores 1987, Reeke & Edelman 1988). Néanmoins elles ne nous satisfont pas complètement, parce qu'elles n'offrent aucune piste pour aborder le problème de la conception, essentiel pour la robotique. Il est significatif qu'en robotique autonome la réflexion constructiviste n'aie pas suscité le foisonnement de nouvelles approches que nous aurions pu en attendre : les travaux ont presque unanimement suivi la même voie, celle de Brooks, parce que c'est la seule qui aborde réellement les questions de conception.

Pour nous, la raison en est que la réflexion constructiviste courante en IA a rejeté la notion même de représentation[27], fondement de toute conception et sans laquelle le roboticien se sent bien démuni. La représentation devient souvent pour les constructivistes un élément émergent, un épiphénomène commode à remarquer, mais n'est plus un pilier de la cognition comme peuvent l'être les symboles dans le cognitivisme. Citons par exemple Varela, qui, écrit :

> «[…] il n'est pas inutile d'insister sur le fait que [un ensemble de critiques que nous approuvons] est une critique de l'usage de la notion de représentation au coeur des sciences cognitives, puisque *seul un monde prédéfini peut être représenté.* Si le monde dans lequel nous vivons se réalise naturellement plutôt que d'être prédéfini, la notion de représentation ne peut plus dorénavant jouer un rôle aussi central.» (Varela 1988, p. 92, italiques rajoutées)

Nous sommes en désaccord sur ce point, et c'est sans doute là notre originalité par rapport aux inspirations constructivistes de la robotique autonome.

---

[27] Ce rejet est spécifique à l'IA ; en psychologie par exemple le constructivisme de Piaget accorde au contraire une grande place à la notion de représentation.

CHAPITRE 3

INCERTITUDES ET PROBABILITÉS

Far better an approximate answer to the
right question, which is often vague, than
an exact answer to the wrong question,
which can always be made precise.
J. Tukey

Ce chapitre décrit les notions théoriques avec lesquelles nous abordons la gestion de l'incertitude. L'application de ces notions au cycle d'adaptation (et au problème des conditions de validité) sera présentée au chapitre suivant.

Nous quittons donc temporairement le cadre robotique pour décrire une théorie du raisonnement probabiliste dûe au physicien américain E.T. Jaynes (1994), que ce dernier a nommée “Probability as Logic” — abréviée PaL par la suite. Cette théorie présente les probabilités comme une extension de la logique booléenne, et formalise les notions d'inférence et d'incertitude. PaL servira de support à la description de toutes nos expériences aux chapitres 4 et 5.

Ce chapitre est organisé ainsi :

La section A présente et justifie l'utilisation des probabilités comme théorie du raisonnement et extension de la logique booléenne.

La section B complète cette présentation avec le principe du maximum d'entropie, qui est l'outil de base permettant d'exprimer sous forme probabiliste une connaissance de nature initialement non probabiliste.

PaL ne manipule que des “états de connaissance”. Or, en robotique, il ne suffit pas de savoir décrire un état de connaissance, encore faut-il l'utiliser pour agir, *prendre des décisions* au vu des connaissances disponibles. La section C discute brièvement de cette question de la décision.

La présentation de PaL est terminée avec cette section C, mais nous ajoutons cependant deux sections supplémentaires : la section D pour présenter le vocabulaire et l'optique avec lesquels nous appliquerons PaL à des problèmes de robotique, et la section E pour discuter des différences entre PaL et la théorie des possibilités (Dubois & Prade 1988), fondée sur la logique floue, et qui est un formalisme alternatif pour les notions d'inférence et d'incertitude.

A. LOGIQUE ET PROBABILITÉS

La logique dont nous parlons est exclusivement la logique classique à deux états. Les exemples utilisés dans cette section sont empruntés à Jaynes (1994).

A.1. Le théorème de Cox

Le théorème de Cox (1946) est le résultat fondamental montrant comment la notion intuitive de plausibilité se formalise par la notion mathématique de probabilité.

Plausibilité d'une proposition logique

Une proposition logique est un énoncé qui est soit vrai soit faux. Si cet énoncé est exprimé en langage naturel, sa véracité et sa fausseté doivent être définies de façon non ambiguë. “Beethoven et Berlioz ne se sont jamais serré la main”, par exemple, peut être considéré comme une proposition logique. En revanche, l'énoncé “la musique de Berlioz est l'égale de celle de Beethoven”, bien qu'on puisse le considérer vrai ou faux, a un sens ambigu. À moins d'avoir formalisé sans ambiguïté ce que signifie pour une musique d'être l'égale d'une autre, nous ne considèrerons pas cet énoncé comme une proposition logique.

Dans bien des cas, étant donnée une proposition logique, les informations disponibles ne permettent pas de trancher avec une certitude absolue entre les réponses “vrai” et “faux”. Nous définissons alors la “plausibilité” d'une proposition comme le degré de certitude que nous avons dans sa véracité. Cette plausibilité dépend évidemment des connaissances mises en jeu pour l'établir : un spécialiste en histoire de la musique aura de nombreux éléments de réponses permettant d'opter nettement pour le fait que Berlioz et Beethoven ne se soient jamais serré la main, tandis qu'une personne n'ayant jamais entendu parler de Berlioz et Beethoven ne saura guère trancher sur cette question.

Notations : étant donné deux propositions logique A et B, nous notons AB leur conjonction, A+B leur disjonction, $\neg A$ et $\neg B$ leurs négations, et l'implication $A \Rightarrow B$ est équivalente formellement à la proposition $(\neg A+B)$. Ces notations sont habituelles en logique booléenne.

Une notion mathématique de plausibilité

Une formalisation de la notion de plausibilité a été proposée par Cox (1946). Notons $\pi(A|C)$ la plausibilité d'une proposition A jugée au vu d'un ensemble de connaissances C. Les hypothèses suivantes (exprimées d'après Jaynes, 1994) définissent alors mathématiquement cette notion :

- Les plausibilités sont représentées par des nombres réels (i.e. $\pi(A|C)$ est un réel). Par commodité, nous conviendrons de représenter une plus grande plausibilité par un nombre plus grand.
- S'il existe plusieurs raisonnements corrects pour déterminer une plausibilité, ils doivent tous mener au même résultat (propriété dite de “consistence”).
- Si de nouvelles informations C’ viennent remplacer les informations C de façon à augmenter la plausibilité de A, alors la plausibilité de la proposition contraire $\neg A$ doit diminuer. Formellement :

  si $\pi(A|C') > \pi(A|C)$, alors $\pi(\neg A|C') < \pi(\neg A|C)$.
- Si de nouvelles informations augmentent la plausibilité de A mais ne concernent en rien une autre proposition B, alors la plausibilité de la conjonction de A et B ne peut qu'augmenter. Formellement :

  si $\begin{cases} \pi(A|C') > \pi(A|C) \\ \pi(B|AC') = \pi(B|AC) \end{cases}$, alors $\pi(AB|C') \geq \pi(AB|C)$
- La dernière hypothèse est une notion de continuité qui ne peut s'exprimer formellement que dans le corps de la démonstration. Intuitivement, une

augmentation infinitésimale de la plausibilité doit se traduire par une aumentation infinitésimale de la valeur numérique qui la représente.

Résultat

On montre alors (Cox 1946, repris dans Jaynes 1994, démontration généralisée par Aczèl 1966) que π vérifie les axiomes précédents si et seulement s'il existe une fonction f telle que :

$$[1] \quad \begin{cases} f \text{ est continue croissante et prend ses valeurs entre 0 et 1} \\ f(\pi(AB|C)) = f(\pi(A|C))\, f(\pi(B|AC)) = f(\pi(B|C))\, f(\pi(A|BC)) \\ f(\pi(A|C)) + f(\pi(\neg A|C)) = 1 \end{cases}$$

Ajoutons alors deux autres hypothèses sur la façon dont doivent être utilisées les connaissances C dans le calcul des plausibilités (il ne s'agit pas vraiment d'hypothèses nouvelles nécessaires, étant des conséquences de celle de consistence déjà adoptée plus haut) :

- Tout raisonnement correct doit prendre en compte l'intégralité des connaissances disponibles, et elles seules.
- Deux problèmes qui ne diffèrent que par l'étiquetage des propositions mènent aux mêmes assignations de plausibilités.

On montre alors que $f(\pi(A|C))$ est déterminée de manière unique par les connaissances C. Ce n'est pas π qui est contrainte par C, mais la composée $f \circ \pi$, qui est unique et caractérise C, et qui est donc l'objet mathématique pertinent pour manipuler des plausibilités. Il y a donc une infinité de solutions pour π, dépendant du choix de la fonction f, qui peut être arbitraire. Le choix le plus simple est de prendre pour f la fonction identité. Ce choix est de même nature que celui du nombre *e* comme base des logarithmes naturels ou des degrés Kelvin pour mesurer les températures, c'est celui qui permet d'écrire le plus simplement les règles [1]. En effet, ces règles deviennent :

$$[2] \quad \begin{cases} \pi \text{ est continue et prend ses valeurs entre 0 et 1} \\ \pi(AB|C)=\pi(A|C)\pi(B|AC)=\pi(B|C)\pi(A|BC) \\ \pi(A|C)+\pi(\neg A|C)=1 \end{cases}$$

Les règles [2] sont une base axiomatique qui définissent un objet mathématique déjà connu : les probabilités). Le théorème de Cox montre ainsi que si l'on admet la notion de plausibilité telle qu'elle a été définie, l'unique façon de manipuler cette notion est donnée par la théorie des probabilités.

Désormais, nous adopterons au lieu de $\pi(A|C)$ la notation habituelle $p(A|C)$, et les règles [2] deviennent alors :

$$[2'] \quad \begin{cases} p \text{ est continue et prend ses valeurs entre 0 et 1} \\ \textit{Règle du produit :} \quad p(AB|C)=p(A|C)p(B|AC)=p(B|C)p(A|BC) \\ \textit{Règle de la somme :} \quad p(A|C)+p(\neg A|C)=1 \end{cases}$$

À propos du terme “probabilité”

La notion de probabilité est plus couramment associée à la notion de fréquence qu'à celle de plausibilité. L'approche dite “fréquentiste” des probabilités considère celles-ci comme une grandeur physique particulière, que la théorie des probabilités permet de manipuler et que les statistiques permettent de mesurer. Le courant dans lequel se place PaL, qui considère les probabilités comme représentant un état de connaissance, est dit “subjectiviste”. Une longue polémique oppose les deux courants, sur le thème “quelle est la bonne approche des probabilités?”. Cette polémique reste très actuelle et virulente[1], et nous souhaitons l'éviter dans le cadre de cette thèse.

Pour cela nous considérons ici les probabilités non comme un objet conceptuel mais comme un objet mathématique, qui donc se définit uniquement par ses manipulations formelles, indépendemment de toute relation à un monde extérieur.[2] Le résultat de Cox dit que la notion conceptuelle de plausibilité se formalise en utilisant la notion mathématique de probabilité.

L'ambiguïté sur le mot “probabilité” n'apparaît alors que lorsqu'on utilise des résultats mathématiques dépendant de la signification effective que l'on attribue à ce mot. En l'occurence, ce qui est spécifique à l'approche fréquentiste des probabilités, ce sont les statistiques. Nous espérons éviter la polémique tant que nous en resterons aux résultats de la théorie élémentaire des probabilités et ne ferons appel à aucun résultat des statistiques[3], et donc tant que nous distinguerons bien les notions de plausibilité et de probabilité.

A.2. De la logique déductive à la logique probabiliste

La logique déductive est un cas limite de PaL

La probabilité d'une proposition logique A prend des valeurs de 0 à 1, 1 étant la certitude absolue que A est vraie, 0 la certitude absolue que A est fausse. Ces deux valeurs extrêmes sont celles qui sont considérées par la logique déductive. Si l'on ne considère que des situations de certitude, l'application des règles [2] mène aux résultats de la logique déductive. Cette dernière apparaît alors comme un cas particulier de PaL.

---

1 Le ton polémique est d'ailleurs excessivement présent dans l'oeuvre de Jaynes lui-même.

2 Illustrons cela par une analogie. Le nombre π apparaît dans la formule de Stirling $n! \sim n^n e^{-n} \sqrt{2\pi n}$ ou dans une expression comme $\int_{-\infty}^{+\infty} \exp(-\frac{x^2}{2})\, dx = \sqrt{2\pi}$. Ces formules semblent avoir peu de rapports avec la géométrie. On pourrait s'en étonner, puisque la “signification” première de π semble de nature géométrique. Faut-il alors chercher à ces formules une interprétation géométrique ? Le fait est que π a des propriétés formelles qui le font ainsi apparaître souvent, indépendemment de toute interprétation géométrique. π est devenu un objet mathématique, entré dans l'histoire des mathématiques pour des raisons géométriques mais finalement contingentes. Pour en revenir aux probabilités, il n'est donc pas choquant que jusqu'à un certain point les mêmes objets mathématiques que sont les probabilités soient adéquats pour formaliser les problématiques fréquentiste ou subjectiviste, malgré leurs différences fondamentales de nature.

3 Cela ne sera pas génant pour nos applications élémentaires. Le besoin d'outils mathématiques plus puissants et sophistiqués, toutefois, rendra à terme cette position neutre plus difficilement conservable.

Par exemple, les deux syllogismes $(A, A \Rightarrow B) \rightarrow B$ (modus ponens), et $(\neg B, A \Rightarrow B) \rightarrow \neg A$ (modus tollens) s'expriment, en posant $C \equiv (A \Rightarrow B)$, par les formules probabilistes $p(B|AC)=1$ et $p(A|\neg BC) = 0$. Ces formules se retrouvent à partir des règles du produit et de la somme.

PaL formalise le raisonnement plausible

Ce que PaL permet de faire que ne permet pas la logique à deux états, c'est formaliser le raisonnement qui porte sur la plausibilité des propositions, lorque les informations disponibles ne permettent pas de mener un raisonnement déductif. Soient les deux propositions suivantes :

$A \equiv$ “Il pleut à midi”,

$B \equiv$ “Le ciel est nuageux à midi moins cinq”.

La relation déductive entre ces deux propositions est $A \Rightarrow B$. Mais il apparaît que si le ciel est nuageux, quand bien même cela ne permet pas d'affirmer qu'il va pleuvoir, cela en augmente fortement la plausibilité — pour par exemple nous inciter à prendre un parapluie. Ce genre de raisonnement, impossible à formaliser dans le cadre de la logique classique, s'exprime de façon probabiliste par (toujours avec $C \equiv (A \Rightarrow B)$) :

$p(A|BC) \geq p(A|C)$,

ce que l'on peut appeler un “syllogisme du raisonnement plausible”. Cette formule se retrouve à partir des règles de base [2].

Il n'est pas nécessaire, pour faire du raisonnement inductif, qu'il existe une connection logique entre les propositions. Un exemple frappant est donné par :

$A \equiv$ “Un homme masqué est en train de fracturer de nuit la porte d'une bijouterie”,

$B \equiv$ “Cet homme est en train de commettre un vol”.

Qu'est-ce qui nous permet, au vu de A, de considérer B comme très plausible ? Il ne s'agit pas d'un raisonnement déductif : l'homme *pourrait* n'être que le propriétaire de la bijouterie qui, revenant tard d'un bal masqué, a perdu ses clefs et essaye de rentrer chez lui. Tout ce que l'on peut dire, c'est que dans notre société le comportement A est plus plausiblement observé chez un voleur (B) que chez toute autre personne ; cela nous le savons grâce à notre connaissance “de sens commun”, que nous appellerons C. Le raisonnement inductif est de dire qu'alors le fait de constater A va renforcer la plausibilité que nous accordons à B. Ceci se retrouve également à partir des règles de base, et s'écrit :

si $p(A|BC) > p(A|C)$, alors $p(B|AC) > p(B|C)$.

Nous appelerons “inférence” tout raisonnement qui de certaines connaissances tire certaines conclusions, qu'il soit déductif ou bien inductif (i.e. du type ci-dessus).

A.3. Conventions de notation et formules utiles

Par la suite, “DDP” sera une abbréviation pour “distribution de probabilités”.

Conventions de notation

Les notations probabilistes utilisées ne sont pas standard.

- Les connaissances “d'arrière-plan”, que nous avons jusqu'ici appelées C, seront systématiquement présentes dans la notation p(). Nous n'écrirons ainsi jamais la probabilité d'une proposition A sous la forme p(A), mais nous préciserons toujours quelles connaissances C ont permis d'assigner une DDP à A, soit par la notation p(A|C), soit de préférence par la notation allégée $p_c(A)$. C'est plus loin (§D.2) que nous justifierons ce choix de toujours spécifier C.
- Seule a été définie la probabilité d'une proposition logique. Si dans une situation C nous assignons des probabilités aux valeurs numériques d'une variable discrète x, nous devrions en toute rigueur noter cela sous une forme propositionnelle qui se révèle assez lourde. Si par exemple x est un entier positif, nous devrions définir l'ensemble des propositions $A_n \equiv$ “x=n” pour $n \geq 0$, et considérer les probabilités $p_c(A_n) = f(n)$, f étant une distribution de probabilités discrète sur l'ensemble des entiers positifs.

  En pratique, lorsque cette notation fonctionnelle f ne sera pas nécessaire, nous dénoterons f implicitement par la notation $p_c(x)$. De même, “$p_c(x)$ pour x=4” désignera $f(4)$.

  La convention est donc que le symbole p désigne une valeur de probabilité si en partie gauche du signe conditionnel apparaissent des propositions, et désigne une distribution de probabilités (DDP) si en partie gauche apparaissent des variables. Les propositions seront notées en majuscules et les variables en minuscules.
- Il est commode de pouvoir parler de probabilités sur des variables continues. Une telle variable x est décrite par une “densité de probabilité” f telle que pour tout ensemble Ω adéquat :

  $p_c(\text{“}x \in \Omega\text{”}) = \int_{x\in\Omega} f(x)\, dx$ (une autre façon de définir f étant par $p_c(A_u)=f(u)du$ avec $A_u \equiv$ “$u \leq x \leq u+du$”).

  Les règles de manipulation [2] restent valables pour les densités de probabilité, et c'est très souvent que des formules seront les mêmes dans les deux cas. Nous continuerons par défaut à manipuler des distributions discrètes, en mentionnant éventuellement les résultats analogues dans le cas continu.

  La convention d'écriture devient : si x est une variable continue, alors $p_c(x)$ désigne la densité de probabilité f telle que pour tout ensemble Ω adéquat, $p_c(\text{“}x \in \Omega\text{”}) = \int_{x\in\Omega} f(x)\, dx$.

Formules utiles

La règle du produit permet d'écrire, sous la forme fonctionnelle décrite ci-dessus :

$p_c(x|y)=p_c(x)p_c(y|x)/p_c(y)$.

La contrainte de normalisation $\sum_x p_c(x|y) =1$ permet alors d'exprimer le dénominateur comme une constante de normalisation, d'où la formule finale très utile :

[3] $$p_c(x|y) = \frac{p_c(x)\, p_c(y|x)}{\sum_x p_c(x)\, p_c(y|x)}.$$

Cette formule a fait couler beaucoup d'encre, sous le nom de “formule de Bayes”. Là encore, le cadre de la logique, où se situe PaL, permet d'éviter les polémiques, qui toutes portent sur l'interprétation en termes de causalité des variables situées à droite ou à gauche du signe “|”. Ne portant que sur des propositions logiques — qui ne définissent ni ne dépendent d'aucune notion de causalité —, ces considérations n'ont aucun sens dans notre théorie ; et la formule [3] n'est à voir que comme une transformation mathématique directe des règles [2].

Pour finir, citons deux autres formules très classiques, toujours déduites des règles [2]. La première retrouve la probabilité d'une disjonction :

[4] $p_c(A+B)=p_c(A)+p_c(B)-p_c(AB)$,

La seconde dit que si $K_1,\ldots,K_n$ sont des propositions mutuellement exclusives — c'est-à-dire que si pour $i\neq j$ on a $p_c(K_iK_j)=0$ —, alors :

[5] $$p_c(A|K_1+\ldots+K_n)=\frac{\sum_i p_c(K_i)\,p_c(A|K_i)}{\sum_i p_c(K_i)},\ i=1\ldots n.$$

Souvent les K_i couvrent toutes les possibilités, au sens que $p_c(K_1+\ldots+K_n)=1$, alors

[6] $$p_c(A)=\sum_i p_c(K_i)\,p_c(A|K_i),\ i=1\ldots n.$$

Enfin, définissons l'espérance et la variance d'une DDP $p_c(x)=f(x)$ par, dans le cas discret :

$$E(x)=\sum_x x f(x),\quad \sigma^2(x)=\sum_x (x-E(x))^2 f(x)=E(x^2)-E(x)^2.$$

et dans le cas continu, Ω étant le domaine de variation de x :

$$E(x)=\int_\Omega x f(x)dx,\quad \sigma^2(x)=\int_\Omega (x-E(x))^2 f(x)\,dx=E(x^2)-E(x)^2.$$

B. LE PRINCIPE DU MAXIMUM D'ENTROPIE

Nous avons vu, dans les paragraphes précédents, les règles par lesquelles PaL manipule formellement les probabilités. Mais nous n'avons pour l'instant aucune idée de comment une connaissance C donnée peut se formaliser en termes probabilistes. Les calculs ne peuvent s'amorcer sans cela. Une réponse à cette question est donnée par le principe du maximum d'entropie, couramment nommé MaxEnt.

Le problème est le suivant : étant donné un ensemble de propositions logiques, nous voulons leur assigner des probabilités, dont la somme fasse 1, et qui satisfassent certaines contraintes C représentant une forme de connaissance. Par exemple, nous sommes un samedi matin devant un distributeur de café, et nous voulons prédire la somme d'argent x qui s'y trouve. Nous savons juste que le préposé passe tous les mardis, et que le montant de son dernier relevé était de 152 francs. Étant donné ces informations, quelle probabilité attribuer à chaque valeur possible de x ?

Il existe une infinité de DDP qui soient compatibles avec C. Toutefois, il n'en existe qu'une qui exploite toute l'information présente dans C et n'en assume aucune autre, dans le sens que tous les paramètres décrivant mathématiquement C sont nécessaires et suffisants pour décrire cette DDP. Cette unique DDP peut être caractérisée mathématiquement par le fait qu'elle maximise, sous les contraintes C, une fonction appelée “entropie”. On l'appelle “distribution de maximum d'entropie”.

Pour une DDP discrète qui à n propositions assigne les probabilités $p_1\ldots p_n$, l'entropie est égale à $H(p_1\ldots p_n)=-\sum_i p_i \log p_i$. Cette définition se généralise au cas

dénombrable ($n \to \infty$), et au cas continu (si f est une densité de probabilité, $H(f) = -\int_x \log f(x)\, dx$).

Intuitivement, l'entropie mesure la “quantité d'information” (au sens de Shannon, 1949) d'une DDP, et la distribution de maximum d'entropie est celle qui minimise l'information non contrainte par la connaissance C.

Ce problème général d'optimisation sous contraintes n'est résolu analytiquement que dans des cas très particuliers, mais heureusement fréquents dans la pratique. Nous commencerons par le cas où nous n'avons sur une variable aucune information explicite autre que les valeurs qu'elle peut prendre. Nous exposerons ensuite le cas le plus général qui ait été résolu analytiquement. Nous insisterons enfin sur le cas extrèmement utile des distributions gaussiennes.

B.1. Le principe d'indifférence

Le principe d'indifférence est un cas particulier du principe de MaxEnt, et en est déductible mathématiquement. Il peut néanmoins être présenté plus intuitivement sans cet intermédiaire.

Soit une variable x, sur laquelle notre connaissance C se réduit aux valeurs qu'elle est susceptible de prendre. Par exemple, C ≡ “x peut prendre des valeurs entières de 1 à 10”. En l'absence de toute information permettant de considérer qu'une valeur est plus plausible qu'une autre, il semble raisonnable de prendre des plausibilités égales pour les 10 valeurs. Les probabilités qui les représentent doivent donc être égales, et puisque toutes les possibilités sont épuisées par ces dix valeurs mutuellement exclusives, la somme de ces probabilités doit être 1. On obtient alors la même probabilité 1/10 pour chaque possibilité :

$p(x=1|C) = \ldots = p(x=10|C) = 1/10.$

De manière générale, en cas d'ignorance totale de la signification d'une grandeur x, n'ayant aucune raison de privilégier a-priori une valeur plutôt qu'une autre, nous lui assignerons a-priori une DDP équiprobable (dans le cas d'une variable continue, c'est la densité de probabilité qui est uniforme).

Si la variable n'est pas bornée, nous ferons comme si elle était bornée par une valeur M, et après avoir mené les calculs nous passerons à la limite $M \to \infty$. Cette précision est importante : le principe d'indifférence n'est pas défini sur des ensembles non bornés, et de tels cas seront gérés exclusivement par un passage à la limite des résultats du cas borné.[4]

À noter encore l'application du principe d'indifférence au cas où les variables sont liées par des contraintes semi-algébriques. Par exemple, si l'on sait que x et y sont des entiers entre 0 et 9 et qu'ils vérifient $x<y$, alors cette contrainte ne fait qu'éliminer des possibilités : sur 100 possibilités pour le couple (x,y), seules 45 sont compatibles

---

[4] Cela peut paraître lourd. Mais les problèmes de manipulation *directe* de quantités infinies sont souvent très délicats à prendre en compte. Jaynes (1994) insiste très fortement sur ce point, et décrit certains résultats très sérieux, considérés par certains comme des “paradoxes” des probabilités (e.g. Kadane & al. 1986, Stone 1970, Dawid & al. 1973), et qui proviennent de pièges mathématiques, sur la manipulation de l'infini, dans lesquelles les auteurs sont tombés. Jaynes affirme que la seule façon claire d'être sûr de son fait, c'est de toujours manipuler des grandeurs finies, puis de passer à la limite une fois obtenu le résultat général. Cette affirmation peut sans doute se discuter, mais en attendant nous préférons suivre ce conseil à la lettre.

avec la contrainte x<y. Le principe d'indifférence assigne alors à chacune la probabilité 1/45, et la probabilité 0 pour les 55 solutions éliminées.

Notons que dans ce qui précède, nous ne disposons d'aucune donnée expérimentale. Le principe d'indifférence est justement utilisé pour assigner des probabilités en l'absence de données expérimentales ; de telles DDP sont qualifiées de “distributions a-priori”.

Le principe d'indifférence a soulevé des controverses qui se résument globalement aux arguments suivants :

- Supposons que x puisse prendre des valeurs de 1 à 6, et que l'on ne sache rien d'autre. Le principe d'indifférence assigne à chaque valeur une probabilité de 1/6. À présent, nous apprenons qu'il s'agit du résultat du lancer d'un dé idéal, c'est à dire totalement symétrique de forme et de poids, et lancé de façon à ne privilégier aucune face. Pour un tel dé nous assignerons à chaque valeur une probabilité de 1/6, pour des raisons de symétrie. On voit alors qu'une ignorance totale et une connaissance totale de cette expérience conduisent au même modèle, ce qui intuitivement est choquant.
- Si on fait un changement de variable $y=x^2$, ou y=log x, ou y=sin x, et que l'on applique le principe d'indifférence à y, cela donne des distributions différentes (notamment dans le cas de densités de probabilité). Ne sachant pas quelle est la “bonne” variable x ou y à laquelle l'appliquer, ce principe peut donner n'importe quoi et est donc inutilisable.
- Dans la pratique, le principe d'indifférence n'est souvent utilisé que pour initialiser des calculs statistiques, et dès que suffisamment de données sont accumulées l'influence du choix initial (des distributions a-priori) devient négligeable. C'est donc se compliquer inutilement la vie que chercher un fondement théorique propre pour déterminer les distributions a-priori initiales, puisqu'en pratique celles-ci ne font aucune différence sur les résultats finaux.

Les réponses à ces arguments sont les suivantes :

- La grande différence entre la connaissance “x peut prendre les valeurs 1 à 6” et la connaissance “x est le résultat du lancer idéal d'un dé parfait”, c'est la façon dont notre connaissance peut évoluer d'après les expériences que l'on fait. Dans le premier cas, la distribution changera avec chaque résultat obtenu, tandis que dans le second cas elle restera stable. Par exemple, observer comme résultat quatre 2 de suite, si l'on sait qu'il s'agit d'un dé parfait, sera considéré comme une simple coïncidence mais n'amènera pas à réviser nos prédictions de 1/6 pour les lancers suivants, tandis que si l'a-priori initial était dû au principe d'indifférence, notre prédiction pour le lancer suivant se sera fortement écartée de l'équiprobabilité pour privilégier le 2.[5]
- Dans tout modèle, le choix des variables pertinentes conditionne la modélisation, et l'argument “pourquoi n'applique-t-on pas le principe d'indifférence à log x plutôt

[5] Si cette différence est souvent ignorée, c'est selon nous à cause du choix universel du jeu de dé comme exemple d'application des probabilité. En effet, la situation est plutôt particulière : la *définition* d'un dé parfait, c'est justement l'équiprobabilité de ses faces, l'impossibilité théorique de découvrir un élément qui en privilégierait certaines. La concordance des distributions de totale ignorance et de totale connaissance n'est pas étrange, car c'est en fait ce que l'on recherche.

qu'à x” s'applique à toute modélisation mathématique. Par exemple, pourquoi modéliser la taille de la population française par une gaussienne, et non le log de cette taille ou son sinus ? Le modélisateur choisit de considérer une variable parce qu'il la juge pertinente au vu de sa propre connaissance tant scientifique que de sens commun, et ceci préalablement à l'élaboration des détails de *tout* modèle.

- Il est vrai que, souvent, l'influence du choix initial des a-prioris devient vite numériquement négligeable devant celle des données expérimentales. Les statisticiens, qui ne travaillent que sur des populations importantes ou de grosses masses de données, peuvent ignorer ce choix initial. Cependant, dans une optique *théorique,* ce principe est fondamental pour obtenir un formalisme consistant et mathématiquement exact même lorsque peu de données sont disponibles. *Ne pas le considérer en pratique est un choix de nature algorithmique* (organiser les calculs en vue d'une application). D'autre part, contrairement aux applications statistiques nos applications robotiques voient souvent comme important le fait de pouvoir raisonner sur des données peu nombreuses, car souvent un robot doit agir au mieux dans l'instant même si ses données ne sont pas pléthoriques. Nous ne pouvons donc pas toujours nous contenter des commodités adoptées par les statisticiens.

B.2. Le résultat principal du principe du maximum d'entropie

Le principe de MaxEnt, théoriquement, permet de déterminer complètement une DDP p(x|C) sur laquelle on connaît certaines contraintes C, l'indétermination étant levée par la maximisation de l'entropie sur toutes les solutions possibles. Il s'agit d'un problème de maximisation sous contraintes souvent insoluble analytiquement. Il existe néanmoins un type de contrainte générique — et fréquent dans la pratique — pour lequel il a été résolu. Il s'agit, pour une variable x, du cas où la connaissance C spécifie que les espérances d'un certain nombre de fonctions de x ont des valeurs données.

Le résultat général (donné pour une distribution discrète) est le suivant :

Soit une variable x pouvant prendre n valeurs discrètes dans un ensemble Ω. On cherche à assigner à x une DDP q(x). Tout ce que nous savons sur x, en dehors des valeurs qu'elle peut prendre, c'est qu'un certain nombre m<n de fonctions $f_1 \ldots f_m$, ont pour espérances respectives des nombres connus $F_1 \ldots F_m$:

[7] $$\sum_{x \in \Omega} f_i(x)\, q(x) = F_i,\ i = 1 \ldots m.$$

De telles fonctions $f_1 \ldots f_m$ sont appelées des “observables”.

La solution q_o du problème de maximisation de l'entropie est alors donnée par la formule suivante :

$$q_0(x) = \frac{1}{Z(\lambda_1, \ldots, \lambda_m)}\, e^{-\lambda_1 f_1(x) - \ldots - \lambda_m f_m(x)},$$

avec $$Z(\lambda_1, \ldots, \lambda_m) = \sum_{x \in \Omega} e^{-\lambda_1 f_1(x) - \ldots - \lambda_m f_m(x)}.$$

Les nombres $\lambda_1 \ldots \lambda_m$ sont des paramètres libres (multiplicateurs de Lagrange), que l'on doit déterminer en résolvant le système d'équations [7] dans lequel q(x) est remplacé par la formule ci-dessus.

Notons de plus que la solution q_o est extrèmement marquée : le “pic” d'entropie est très “pointu” (théorèmes de concentration de l'entropie de Robert (1990) et Jaynes (1982)).

Donnons à présent un exemple très fréquemment rencontré : celui d'une variable entière ou réelle pour laquelle les observables sont $f_1(x)=x$ et $f_2(x)=x^2$, c'est à dire que les valeurs contraintes sont l'espérance et la variance de la distribution cherchée.

B.3. Un cas particulier utile : les distributions gaussiennes

x est une variable sur laquelle la seule connaissance disponible est :

$$C \equiv \begin{cases} x \text{ prend des valeurs entières} \\ ; \text{ elle a pour espérance } E(x)=\mu \\ ; \text{ elle a pour variance } \sigma^2(x)=\sigma^2. \end{cases}$$

Alors la distribution de maximum d'entropie est la distribution gaussienne de paramètres μ et σ :

[8] $$p_c(x|\mu\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Cette fonction sera par la suite notée $G(x|\mu,\sigma)$. La barre “|” n'a pas la même signification que dans la notation p() : elle sépare ici les variables des paramètres, alors que la notation p() concerne en définitive toujours des propositions, et ne fait aucune telle distinction. La confusion entre ces notations n'est pas gênante en pratique, elle rend plutôt les formules plus claires.

Le cas d'une variable réelle donne ce même résultat [8] en tant que densité de probabilités.

Dire que la gaussienne est la distribution de maximum d'entropie pour les observables x et x^2 équivaut à dire que c'est la seule DDP qui soit totalement déterminée par la connaissance de ses deux premiers moment.

Ce résultat est, actuellement, la seule conséquence pratique du principe de MaxEnt dans nos travaux (avec le principe d'indifférence). Ce principe a eu pour nous un intérêt essentiellement théorique jusqu'à présent, mais nous comptons l'exploiter plus par la suite (notons à ce propos que l'application pratique du résultat donné au paragraphe précédent nécessite des techniques de résolution numérique).

C. DÉCISION ET ACTION

Imaginons un robot mobile dont l'état de connaissance probabiliste comprenne les variables a et d, respectivement azimut et distance de l'obstacle le plus proche. C désigne un ensemble de connaissances et données permettant de les calculer sous la forme p(a|C) et p(d|C). Supposons que p(d|C) soit une gaussienne d'espérance 30cm et écart-type 10cm, et que p(a|C) soit une distribution bimodale avec un pic autour de -1° et un pic autour de +5°. La question est : que doit faire ce robot ? Doit-il agir “comme si” la distance était de 30cm ? doit-il tourner vers la gauche ou la droite, et doit-il ralentir ? et ne devrait-il pas inclure une part d'aléatoire dans ces choix ? Il est clair que la connaissance de a et d (via une DDP) ne suffit pas à déterminer une

action. Un état de connaissance probabiliste est purement descriptif ; pour agir, choisir une valeur déterminée, d'autres critères sont nécessaires.

Les diverses réponses à ce problème de décision (revue dans Jaynes 1994 et Pearl 1991) partagent la même idée de base : on adjoint au pur état de connaissance probabiliste des “fonctions d'utilité” (ou “de coût”), qui représentent l'intérêt, le coût ou le danger potentiel des décisions possibles. On choisit alors l'action maximisant l'utilité espérée (ou minimisant le coût) à partir de la connaissance probabiliste disponible.

Très simplement, imaginons que l'on aie le choix entre deux actions A_1 et A_2. Les conséquences possibles de ces actions sont B_1, B_2 et B_3. À chacune de ces conséquences on associe les utilités $U(B_1)$, $U(B_2)$ et $U(B_3)$. L'utilité d'une action A_i est alors définie comme l'utilité espérée (somme pondérée) de ses conséquences étant donné les connaissances C disponibles :

$$U(A_i|C) = U(B_1)\, p(B_1|A_iC) + U(B_2)\, p(B_2|A_iC) + U(B_3)\, p(B_3|A_iC).$$

On choisit l'action dont l'utilité est la plus grande. Ce cadre simple est celui de Pearl (1991), mais d'autres formulation sont possibles pour les fonctions d'utilité, par exemple en termes d'estimation de paramètres (théorie de la décision de Wald, décrite par Jaynes).

Dans nos travaux, cette formalisation des processus de décision n'est pas développée. Nous nous intéressons principalement à l'inférence et la mise à jour des états de connaissance. L'utilisation de ces connaissances pour l'action est pour l'instant laissée (comme nous le verrons) à l'appréciation du programmeur.[6]

D. APPLICATION À LA ROBOTIQUE

Nous allons utiliser PaL dans le cadre particulier de la robotique autonome. Pour cela, nous allons utiliser certains termes spécifiques que nous définissons au §D.1. Ce cadre impose de plus un biais dans la façon de poser les problèmes : il faut les poser sous des formes les plus systématiques possibles, plutôt que chercher des ruses permettant une résolution plus aisée mais moins systématique. Cette optique est précisée au §D.2.

D.1. Terminologie

Sachant que dans cette thèse la connaissance C que nous voudrons représenter sous forme probabiliste sera toujours celle d'un robot, il est utile de classifier les différentes composantes de C selon cette application particulière. Nous distinguerons ainsi trois types de connaissances :

- Au cours de son activité, un robot muni de capteurs et d'actionneurs pourra observer son interaction avec l'environnement par l'intermédiaire des valeurs prises au cours du temps par ces capteurs et actionneurs. Ces données empiriques constituent les “informations expérimentales”.

---

[6] Une approche théorique de la décision sera certainement nécessaire un jour ou l'autre, mais nous ne pensons pas qu'elle doive forcément s'exprimer en termes de coûts ou d'utilité. Nous retardons donc l'attaque de cet aspect décisionnel en attendant une autre piste à suivre. Nous pensons, mais ce n'est qu'une pure spéculation pour l'instant, que la notion de clôture opérationnelle (Varela 1989) constitue l'embryon d'une telle piste.

- Pour que le robot tienne compte de ces informations expérimentales, il faut qu'il sache comment le faire. Supposons qu'un capteur délivre les valeurs 98, 125, 131, 142, 166, 162… En l'absence d'autres informations, le robot n'a pas de moyen de modifier son comportement en réponse à ces données observées. Il faut d'une façon ou d'une autre représenter le fait que par exemple il s'agit d'un signal photoélectrique, et que cette connaissance permet de décider un mouvement de suivi de lumière, ou bien permet d'apprendre un paramètre intéressant. Une connaissance de ce type est dite “procédurale”.
- Au moment où le robot est mis en marche, il ne dispose pas encore de données expérimentales. Pourtant, il va devoir agir. S'il est commandé par un système probabiliste, il va falloir donner au système les connaissances nécessaires à sa mise en route initiale. Ces connaissances sont dites “a-priori”.

L'ensemble des connaissances a-priori et procédurales est appelé “connaissances préalables”.[7] Ce sont celles qui devront être présentes au lancement du robot — et qui devront donc avoir été élaborées par le concepteur.[8]

Nous noterons habituellement D les données expérimentales engrangées au cours du temps, et C les connaissances préalables.

Ces connaissances expriment, typiquement, que l'on a appris à prédire certaines variables sensorielles x en fonction d'autres variables sensorielles y et des commandes exécutées z. Les connaissances C spécifient donc la forme de $p_c(x|yzD)$. On appelle “problème direct” la spécification de telles distributions.

Les problèmes qu'on se pose alors sont souvent du type : sachant les valeurs de y, et désirant observer certaines valeurs de x, quelles sont les commandes z à exécuter ? La distribution désirée est donc $p_c(z|xyD)$. Il s'agit d'un “problème inverse”, c'est-à-dire qu'à partir des connaissances C il faut pour le résoudre utiliser la formule de Bayes :

[9] $p_c(z|xyD)=Z^{-1}\, p_c(z|yD)\, p_c(x|yzD)$,

où Z est la constante de normalisation $Z = \sum_z p_c(z|yD)\, p_c(x|yzD)$.

Dans cette formule, il faut remarquer que si $p_c(z|xyD)$ et $p_c(z|yD)$ sont des DDP par rapport à la variable z, $p_c(x|yzD)$ est quant à elle une DDP sur x. Cependant, nous considèrerons souvent cette quantité en tant que fonction de z ; ce n'est alors plus une DDP, et on l'appelle la “vraisemblance” de z.

---

[7] Notons que si l'on revient à la distinction structure-paramètres décrite au chapitre 2, dans un système formalisé en termes probabilistes, la structure est représentée par les connaissances préalables et les paramètres sont identifiés à l'aide des informations expérimentales.

[8] Un physicien appellerait ces connaissances préalables des “hypothèses”. De fait, techniquement, ces deux notions ont le même rôle, mais conceptuellement les termes véhiculent des sous-entendus différents. Les connaissances préalables ne constituent pas à proprement parler une hypothèse (à vérifier) sur la façon dont fonctionne un environnement extérieur au système, mais définissent la façon dont fonctionne le système lui-même, de même qu'un programme informatique ne traduit pas une hypothèse faite par le programmeur, mais une connaissance de ce dernier. Nous reviendrons sur ce point important au chapitre 4, §C.3.

D.2. Systématicité des calculs

L'exemple suivant est dû à Jaynes. Pierre et Paul jettent alternativement une pièce de monnaie, et le premier qui obtient “face” gagne le jeu. C'est Pierre qui commence. Quelles sont les probabilités p et q que Pierre ou Paul gagne ? Une solution brutale est de sommer $(1/2)^n$ sur les entiers pairs ou impairs :

$$p = \sum_{n=0}^{\infty} \frac{1}{2^{2n+1}} = \frac{2}{3}, \quad q = \sum_{n=0}^{\infty} \frac{1}{2^{2n}} = \frac{1}{3}.$$

Mais si l'on réfléchit on peut faire bien plus simple : on note que les rôles de Pierre et Paul sont échangés si Pierre perd au premier coup, donc q=p/2 et donc p=2/3 et q=1/3.

Dans une optique de résolution d'un problème particulier, la solution rusée est la bienvenue. Dans notre optique générale et cognitive, la première solution est ce qui nous intéresse, car elle procède d'une méthode systématique et permanente, et ne repose pas sur des “trucs” qu'un humain intelligent peut trouver mais pas une machine.

PaL se prête à notre avis à un traitement systématique. Les contraintes que cela impose sont parfois vues comme des complications inutiles, mais l'exemple ci-dessus illustre bien les questions en jeu. Il est très contraignant de chercher à mettre en place des méthodes de raisonnement qui ne fassent pas appel à des arguments de “sens commun” — mais c'est la condition pour obtenir un système d'inférence totalement automatisé.

C'est bien là la raison fondamentale pour laquelle nous “traînons” toujours explicitement les connaissances préalables C dans les notations $p_C(\ldots)$ ou $p(\ldots|C)$. En effet, si notre objectif était purement “applicatif”, c'est-à-dire de résoudre par des méthodes probabilistes certains problèmes donnés, nous pourrions nous en dispenser, il suffirait de garder ces connaissances à l'esprit implicitement, comptant sur notre bon sens pour ensuite les exploiter intelligemment. Mais notre but ultime est de systématiser les méthodes utilisées, de les implémenter par des processus informatiques totalement automatiques, et non de développer une simple “boîtes à outils mathématique” disponible pour le concepteur.

PaL fait donc *systématiquement* et *explicitement* référence à la connaissance préalable C ayant permis de formaliser un problème. Le modèle obtenu peut alors être décrit informatiquement. Toute carence, tout ajout, et en général tout changement dans les connaissances préalables doit être explicité. Par cette méthode, on évite de plus certains pièges décrits par Jaynes — même les utilisateur humains ont tendance à se guider sur les notations utilisées, et à comparer des probabilités qui ne sont en fait pas comparables parce que résultant de connaissances préalables de natures différentes.

E. AUTRES THÉORIES DU RAISONNEMENT INCERTAIN

À ce stade, il est nécessaire de citer les théories “concurrentes”, qui ont été développées pour aborder le problème de l'incertitude avec la même ambition que dans ce chapitre : par exemple la théorie des possibilités (Dubois & Prade 1988, Dubois & al. 1994a), ou la théorie Dempster-Shafer sur les “belief functions” (pour une discussion générale voir le numéro spécial du “journal of applied non-classical logics”, vol. 1 n°2/1991). Les défenseurs de ces théories affirment qu'elles

permettent d'aborder des problèmes hors de portée des approches probabilistes (Dubois & Prade 1994, Ruspini 1989, Bouchon-Meunier 1989, Dubois & al. 1994b). Typiquement voici ces problèmes pour lesquels une formalisation probabiliste serait mal adaptée :

- Les probabilités ne distinguent pas la notion d'incertitude de celle d'imprécision. L'imprécision concerne la qualité intrinsèque des données disponibles (information d'observation), indépendemment de leur utilisation, et l'incertitude concerne en revanche cette utilisation (information d'exploitation). Les probabilités seraient adéquates pour traiter l'incertitude mais pas l'imprécision.
- Les probabilités subjectives seraient inséparables de la notion de décision, et ne prendraient leur sens qu'en termes de ce qu'on serait prêt à parier sur l'occurence de tel ou tel événement. Or, cette notion est trop pauvre comme approche générale de l'inférence, car en général un état de connaissance ne peut se réduire à un état d'incertitude concernant les actions à entreprendre.
- Les probabilités ne pourraient exprimer l'ignorance car elles ne s'intéresseraient qu'aux prédictions et non à l'incertitude quant à ces prédictions elles-mêmes ; typiquement l'ignorance totale sur un dé mène au même modèle que celui d'un dé idéal parfaitement connu, or on aimerait modéliser cette différence.

Ces critiques sont injustifiées et témoignent d'une confusion entre ce qui est du ressort d'une théorie et ce qui est du ressort de leurs méthodologies d'application. Pour tous les exemples que nous avons rencontrés illustrant les points ci-dessus, l'approche probabiliste était tout à fait adéquate et efficace. Ce ne sont éventuellement que certaines méthodologies qui, adaptées à certains types de problèmes, ne le sont pas à d'autres : par exemple, poser un problème d'inférence en termes de décision n'est pas toujours bien adapté, mais il est injustifié de considérer les problèmes qui en résultent comme intrinsèques à la théorie probabiliste même, alors qu'ils ne proviennent que d'une mauvaise application.

Voici nos réponses aux trois points ci-dessus :

- Incertitudes et imprécision ne sont pas des notions fondamentalement distinctes, mais deux aspects d'un même problème (l'incomplétude). Il peut être commode, sur un plan *méthodologique,* de les distinguer, mais leur traitement unifié sur le plan *théorique* nous semble au contraire extrèmement souhaitable. Chercher à séparer ces deux aspects par la théorie, c'est à notre sens retomber dans l'erreur de confondre le monde avec la connaissance qu'on en a.
- Inférence et décision sont des problèmes séparés. PaL est une théorie de l'inférence, et sa pertinence ne dépend pas de la théorie de la décision qu'on peut vouloir y appliquer. Le fait que certaines méthodes probabilistes, adaptées spécialement pour des problèmes de décision, ne conviennent pas pour d'autres types de problèmes, ne suffit pas à remettre en cause la théorie. Le fait de permettre d'aborder des problèmes de décision autant que d'inférence est en fait une *force* des probabilités.
- Il est nécessaire de distinguer un modèle de ses prédictions. Le modèle d'un dé idéal mène aux mêmes prédictions que celui d'un tirage entre 1 et 6 par un dispositif inconnu, mais les états de connaissance représentés ne sont pas les mêmes. Typiquement, ils ne vont pas réagir de la même façon à l'acquisition d'informations nouvelles (de fait, la concordance des prédictions n'est vraie qu'en

l'absence de données expérimentales, voir §B.1). Les probabilités distinguent parfaitement les deux situations “dé parfait” et “ignorance totale du dispositif utilisé”.

Aucun des problèmes qu'on nous a cités comme exemples n'échappait au pouvoir descriptif des probabilités. Nous reconnaissons sans nul doute qu'il puisse se trouver des théories plus puissantes permettant d'aborder des problèmes nouveaux — mais celles que nous avons rencontrées (e.g. la théorie des possibilités) semblent avoir des avantages plus techniques et algorithmiques (mise en oeuvre plus rapide, sur des problèmes particuliers, que des techniques probabilistes) que théoriques ou conceptuels.

CHAPITRE 4

APPRENTISSAGE

C'est bien une machine qui a gagné la guerre,
John, tout au moins un petit système de calcul très
simple, dont je me sers chaque fois que je dois
prendre une décision particulièrement difficile…
Pile ou face, messieurs ?
Isaac Asimov

Dans ce chapitre nous présentons un cycle d'apprentissage élémentaire, illustré sur des expériences très simples. Notre but est de mettre en oeuvre une gestion de l'incertitude telle que nous l'avions requise au chapitre 2. Nous nous fondons pour cela sur le cadre probabiliste du chapitre 3, et nous présentons la réalisation d'un système d'inférence probabiliste élémentaire.

La section A présente “l'expérience de la bassine lumineuse”, qui consiste à explorer un environnement lumineux avec une cellule photoéléctrique. Cette expérience illustre comment les problèmes de robotique peuvent s'exprimer en termes d'inférence probabiliste.

La section B montre comment nous avons défini et programmé un système d'inférence probabiliste.

La section C présente les résultats de l'expérience, et montre dans quelle mesure la gestion des incertitudes permet d'aborder le problème des conditions de validité.

Le système d'inférence décrit en B suit de près les notions théoriques, mais les calculs effectués risquent de devenir trop complexes pour des problèmes de grande dimension. La section D discute des adaptations que devra subir ce système dans cette perspective, et présente une architecture alternative possible que sont les “réseaux bayésiens” (Pearl 1991).

La section E, enfin, présente une seconde expérience consistant à faire apprendre à un petit robot mobile Khépéra un comportement d'évitement d'obstacles.

A. L'EXPÉRIENCE DE LA BASSINE LUMINEUSE

La bassine lumineuse a été imaginée par Olivier Lebeltel comme le système robotique le plus simple qui puisse illustrer nos réflexions. Ce robot ne dispose que d'une modalité sensorielle (un capteur photoélectrique) et d'une modalité motrice (la commande de position angulaire d'un axe). Néanmoins, les leçons que nous en avons tirées sont d'une richesse étonnante comparée à la simplicité du dispositif.[1]

---

[1] À la suite de certaines critiques concernant cette simplicité, nous tenons à bien insister sur la vocation purement illustrative de cette expérience. *Il ne s'agit pas d'une application* ; nous savons bien qu'il y aurait d'autres façons de traiter ce dispositif ; nous voulons juste montrer comment nous *pouvons* le traiter de façon probabiliste. Conceptuellement, cette expérience est l'équivalente de celle d'évitement d'obstacles du robot mobile Khépéra que nous décrirons plus loin, d'apparence pourtant plus respectable car plus à la mode. La poubelle lumineuse est toutefois plus simple à décrire ; dans ce cadre *la simplicité n'est pas un défaut mais une qualité.*

Par convention, les symboles du type `angle`, `cell`, `env` dénoteront des variables, et les mêmes en italiques (*`angle, cell, env`*) dénoteront des valeurs particulières de ces variables.

A.1. Dispositif expérimental

La bassine lumineuse est une bassine en plastique verte éclairée par une lampe halogène placée sur un bord à mi-hauteur environ. Dans cet environnement lumineux, on place une cellule photo-électrique qui peut tourner autour d'un axe vertical comme schématisé figure 1. La commande a été réalisée sur un bras de robot SCEMI, utilisé également pour d'autres expériences du chapitre suivant. Les détails matériels et informatiques sont reportés à l'annexe 1.

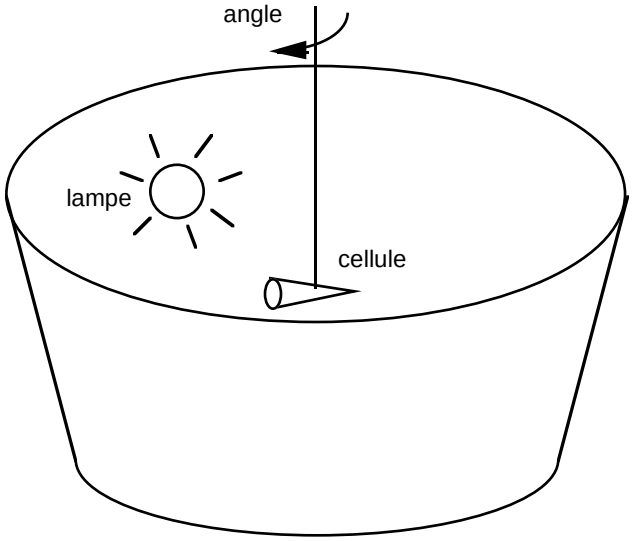


Figure 1. Schéma de la bassine lumineuse : l'axe est commandé en position (angulaire), et la cellule photoélectrique permet d'échantilloner sur presque 360° le “paysage lumineux” produit par la lampe dans la bassine.

L'axe est commandé en position par un angle `angle` défini modulo 360° entre -180° et +180°. Les positions proches de ±180° sont interdites par des butées mécaniques ; concrètement, une commande en dehors de [-178°,178°] n'a aucun effet sur le robot. Nous utiliserons en pratique des commandes discrètes, par exemple tous les 5° de -175° à +175° (ce qui fait 71 valeurs possibles pour `angle`).

La cellule photo-électrique délivre une valeur discréte entre 0 (noir total) et 2047 (saturation). Cette variable sensorielle sera notée `cell`. Les bornes 0 et 2047 sont celles du convertisseur analogique-numérique, elles peuvent se révéler inatteignables en pratique.

Ce dispositif permet d'explorer l'environnement lumineux que constitue la bassine, par l'observation des conséquences sensorielles `cell` de commandes motrices `angle`. Les figures 2a, 2b et 2c montrent une série de mesures réalisées dans trois environnements différents. On peut donner un nom à ces environnements par exemple ξ_1 pour celui de la figure 2a, et ξ_2 et ξ_3 pour ceux des figures 2b et 2c. Nous désignerons par `env` une variable pouvant prendre les valeurs ξ_1, ξ_2, ou ξ_3.

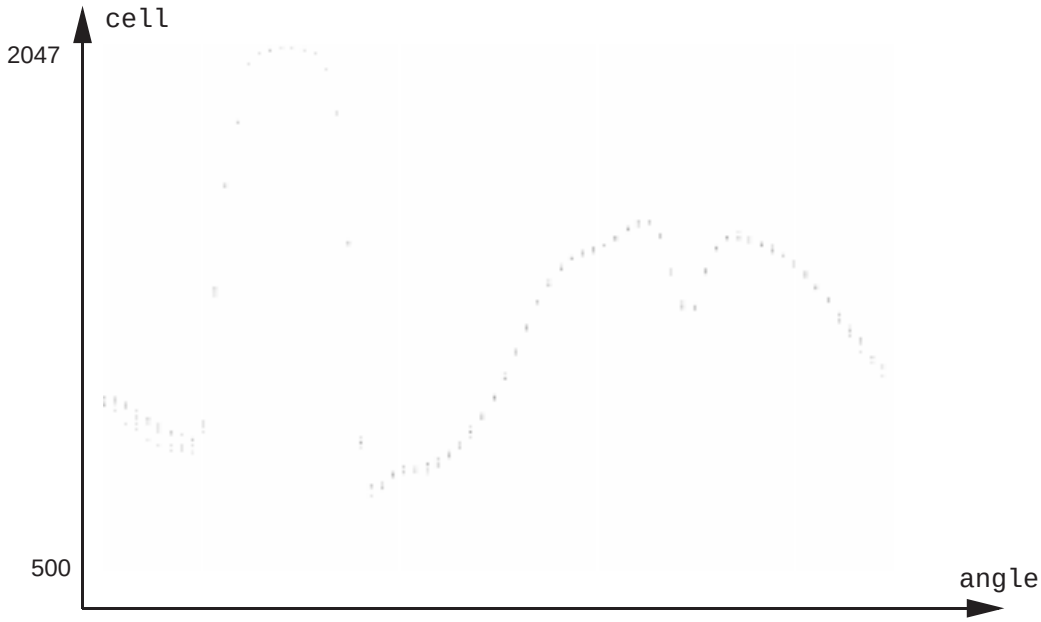


Figure 2a. Les mesures (*angle, cell*) dans l'environnement ξ_1.

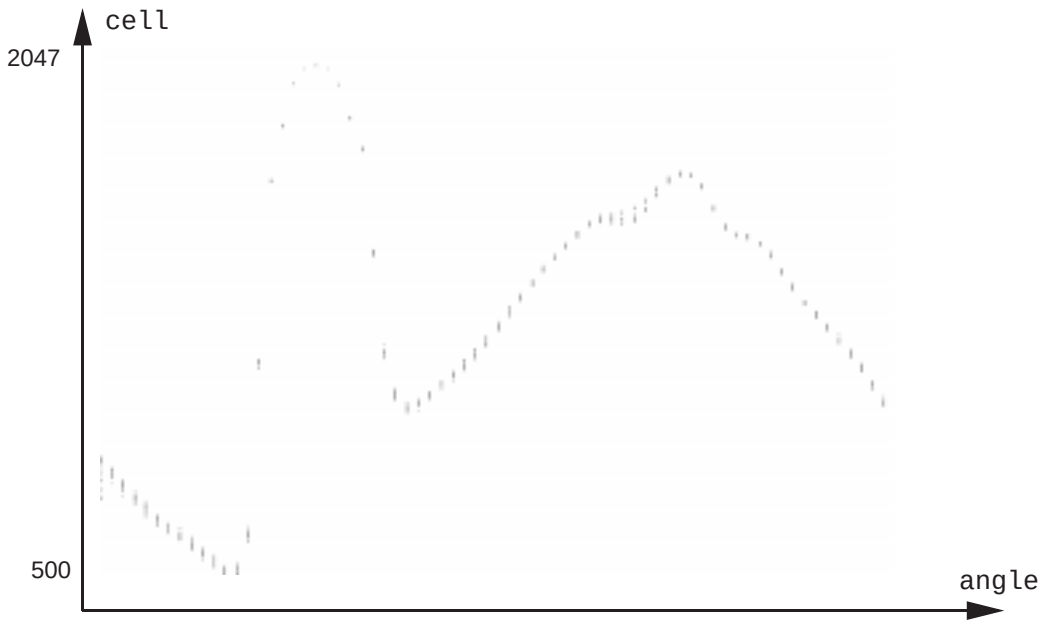


Figure 2b. Les mesures (*angle, cell*) dans l'environnement ξ_2.

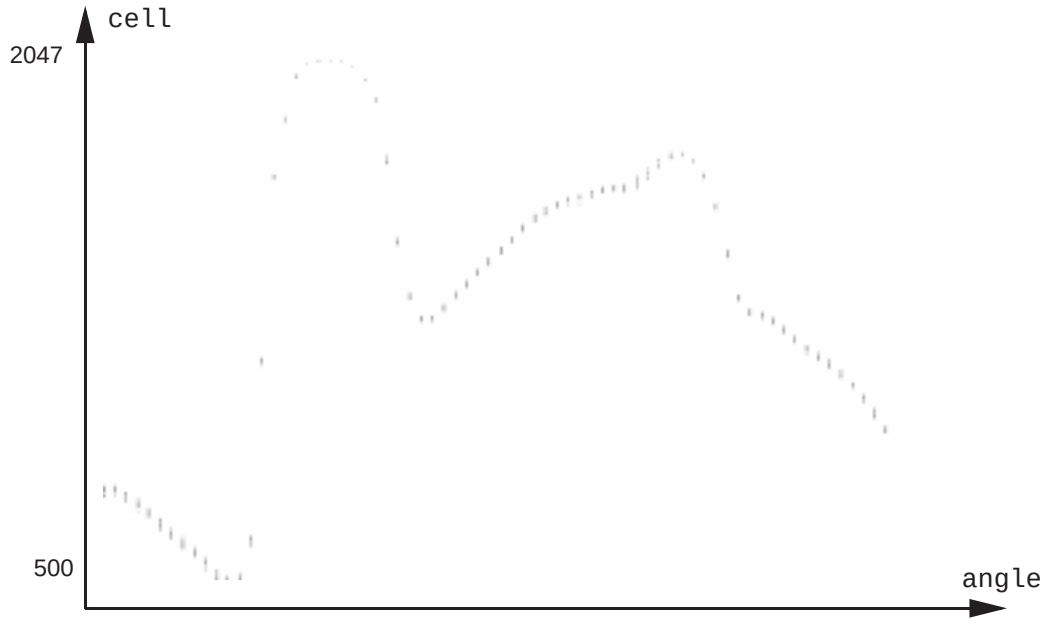


Figure 2c. Les mesures (*angle, cell*) dans l'environnement ξ_3.

Les trois environnements correspondent à des positionnements différents de l'axe dans la bassine. Dans chacun, on voit bien le pic (saturé) correspondant aux positions face à la lampe, et la réflexion principale en face de cette lampe. Cette réflexion est modulée par les ombres complexes de l'axe et de la cellule.

A.2. Modélisation probabiliste

Nous voulons construire une relation probabiliste entre les variables `angle`, `cell` et `env` de la bassine lumineuse. Le raisonnement est élémentaire, mais nous le détaillons néanmoins afin de bien montrer le rôle des différentes formes de connaissances préalables.

L'expérimentation consiste à faire un ensemble de mesures. Chaque mesure consiste, dans un environnement donné *env,* à positionner l'axe à un angle *angle* tiré aléatoirement, et à lire la valeur *cell* délivrée par la cellule. Les données expérimentales, notées `data`, sont donc un ensemble de triplets (*angle, cell, env*), représentés dans notre exemple par tous les points des figures 2a, 2b et 2c.

Le protocole expérimental utilisé pour explorer la bassine est une exploration aléatoire équiprobable : à tout instant, `angle` est déterminée par le tirage équiprobable d'un multiple de 5° dans l'intervalle [-175°,175°] (71 valeurs possibles). Nous avons donc un a-priori p_c(`angle`) constant :

[1] p_c(`angle`)=1/71, pour `angle` $\in$ {-175,170,…,170,175}.

Ce protocole expérimental est fixé, indépendant des données, et reste le même dans tous les environnements. Cela s'exprime par :

[2] p_c (`angle` | `env data`) = p_c (`angle`).

Similairement, le choix de l'environnement `env` est équiprobable et indépendant des données :

[3] p_c (`env` | `data`) = p_c (`env`) = 1/3, pour `env` $\in \{\xi_1,\xi_2,\xi_3\}$.

La connaissance informelle que nous avons du dispositif expérimental et des conditions où nous l'utilisons nous indique que `cell` dépend principalement de `angle` et de `env` (la mesure de la cellule reste à peu près constante pour une commande donnée du capteur dans un environnement donné). Nous traduisons formellement cette connaissance par le fait que les observables considérés lors des expériences sont les deux premiers moments de p_c (`cell` | `angle env`). Le principe de MaxEnt nous dit alors que cette DDP est une gaussienne[2] :

[4] p_c (`cell` | `angle` `env`) = G (`cell` | μ(`angle`, `env`), σ(`angle`, `env`)).

Rappelons que la notation G(x|μ,σ) désigne la fonction de x gaussienne d'espérance μ et d'écart-type σ.

Le passage ici par le principe de MaxEnt peut sembler artificiel. D'une part il est clair que le choix de conception est en fait inverse : nous avons choisi la forme gaussienne, et la justification par MaxEnt est venue a-posteriori. D'autre part, chacun dans cette situation nous accorderait volontiers le droit de faire une hypothèse gaussienne. Nous répondrons à ces arguments un peu plus tard dans cette section — en attendant nous gardons la formulation ci-dessus.

Dans la formule [4], les deux paramètres μ et σ sont des fonctions de `angle` et `env`. Or, nous voulons, en fait, *apprendre* ces paramètres, c'est-à-dire les déterminer à partir des données expérimentales `data`. La formule [4] se réécrit alors plutôt :

[5] p_c (`cell` | `angle` `env` `data`) = G (`cell` | μ(`angle`, `env`, `data`), σ(`angle`, `env`, `data`)).

Le processus d'apprentissage est alors spécifié par les fonctions μ(`angle`, `env`, `data`) et σ(`angle`, `env`, `data`). Le choix est très ouvert, selon que l'on désire des propriétés d'interpolation, une forme finale plus ou moins compacte (au sens du nombre de paramètres nécéssaires), etc… Le plus simple est de définir μ et σ indépendemment les unes des autres pour les 71 valeurs discrètes de `angle` et les 3 valeurs de `env`, c'est-à-dire sans traduire aucune hypothèse de continuité (et donc sans spécifier de propriétés de généralisation).

Nous prenons donc pour μ(`angle`, `env`, `data`) la moyenne expérimentale de toutes les valeurs de `cell` retenues dans les données en fonction de la position `angle` dans l'environnement `env`, et pour σ l'écart-type expérimental de ces valeurs à cette moyenne μ. L'ensemble des valeurs obtenues pour μ et σ est représenté figure 3, pour env = ξ_1.

---

[2] Nous oublions ici une connaissance importante, qui est que `cell` est bornée entre 0 et 2047. La solution exacte serait de renormaliser dans cet intervalle les distributions gaussiennes obtenues. Cette renormalisation est en pratique négligeable, et ce même aux bornes de l'intervalle. En effet, ces bornes (0 et 2047) sont celles du convertisseur analogique-numérique, et les valeurs effectives de saturation sont situées loin de ces bornes ; en fait pour déterminer théoriquement ces valeurs il faudrait connaître le détail du montage analogique en amont de cette conversion. Le montage (décrit en annexe 1) ayant été réglé empiriquement, cette connaissance n'est alors pas disponible avant expérimentation.

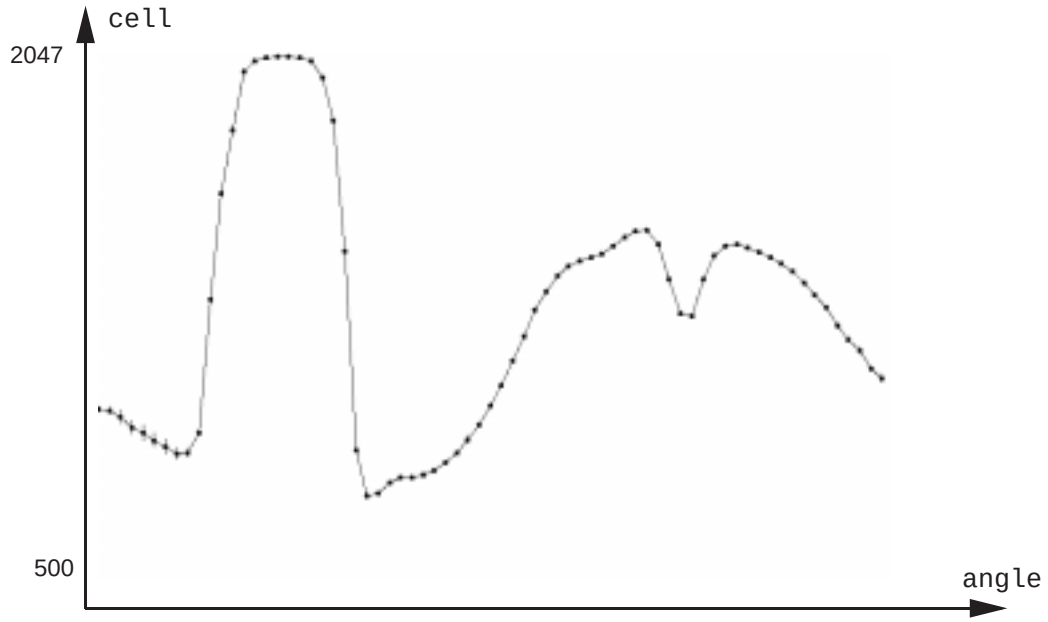


Figure 3. Représentation de p_C (`cell` | `angle env data`) pour l'environnement ξ_1. `cell` est en ordonnée : chaque point représente la moyenne μ obtenue pour une valeur donnée de `angle` en abcisse ; les barres verticales représentent un écart-type σ. Les points sont reliés entre eux pour des raisons esthétiques uniquement, notre modélisation ne prenant en compte aucune notion d'interpolation. Cette façon de représenter des paramètres gaussiens μ et σ sera amplement employée par la suite.

Nous avons fini de formaliser toutes les connaissances préalables que nous avions. Nous pouvons donc à présent expliciter totalement C *formellement* par C ≡ [6] :

[6]
p_C (`angle`) = p_C (`angle` | `env data`) = 1/71, `angle` ∈ {-175,170,…,170,175},
p_C (`env`) = p_C (`env` | `data`) = 1/3, `env` ∈ $\{\xi_1,\xi_2,\xi_3\}$,
p_C (`cell` | `angle env data`) = G (`cell` | μ (`angle`, `env`, `data`), σ (`angle`, `env`, `data`)),
avec μ et σ (`angle`, `env`, `data`) = moyennes et écarts-types expérimentaux.

Cet ensemble de connaissances préalables et les données expérimentales D permettent en théorie de calculer la probabilité de n'importe quelle proposition de type "`angle` ∈ U, `cell` ∈ V, `env` ∈ W" (U, V et W étant des ensembles convenables). En effet, elle permet de calculer la DDP conjointe p_C (`angle cell env` | `data`) = p_C (`angle`) p_C (`env`) p_C (`cell` | `angle env data`), à partir de laquelle on peut par sommation de `angle`, `cell` et `env` sur U×V×W calculer la probabilité désirée.

A.3. Inférences réalisables sur le modèle probabiliste

Nous pouvons conduire sur C toutes sortes d'inférences :

- p_C (`cell` | `angle env data`) permet de prédire la valeur que l'on lira après avoir positionné la cellule dans une position donnée, connaissant l'environnement. C'est ici une "inférence directe", c'est-à-dire ne faisant pas appel à la formule de Bayes.

- p_c(`angle` | `cell env data`) permet de prédire la ou les positions les plus susceptibles d'amener la cellule à lire une valeur donnée. Le calcul de cette DDP fait appel à la formule de Bayes, et on l'appelle “inférence inverse”. Par exemple, la figure 4 montre deux distributions p_c(`angle` | `cell env data`) obtenues pour `cell`=750 et 1250 (avec `env`=ξ_1). On voit diverses solutions possibles pour `angle` : prendre la valeur la plus probable de cette DDP, ou son espérance, ou tirer la valeur aléatoirement selon la loi p_c(`angle` | `cell env data`). Le choix d'une méthode particulière est un problème séparé de celui de l'inférence.
- p_c(`env` | `angle cell data`) permet au vu d'une mesure expérimentale (*`angle`*, *`cell`*) de prédire l'environnement actuel.
- p_c(`cell` | `angle data`) permet de prédire la valeur lue sur la cellule à la position `angle` dans l'ignorance de l'environnement.

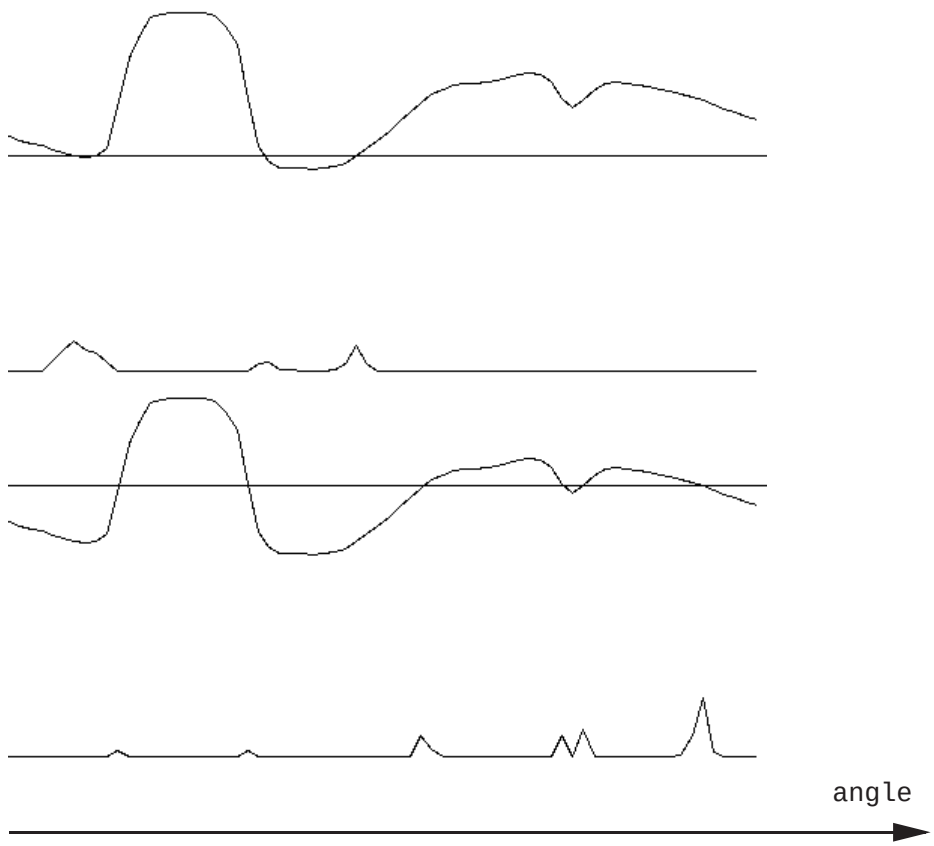


Figure 4. Deux distributions inverses p_c(`angle` | `cell` `env` `data`) pour `env`=ξ_1. La première ligne représente la courbe μ(`angle`) pour ξ_1, coupée par la droite `cell`=750. La seconde montre en regard p_c(`angle` | `cell`=750 `env`=ξ_1 `data`), c'est-à-dire les probabilités de chaque position ou peut se trouver la cellule sachant l'intensité lue. Les troisième et quatrième lignes montrent la même chose pour `cell`=1250.

En robotique, les problèmes les plus ardus sont toujours les problèmes “inverses” du type “inférer `angle` connaissant `cell` et `env`”, car ce sont très souvent des problèmes mal posés, où il s'agit d'inverser des fonctions qui ne sont pas toujours inversibles. Un modèle probabiliste a de ce point de vue un avantage énorme sur un

modèle fonctionnel : une DDP est toujours inversible (par la formule de Bayes), et donc l'inversion n'est pas un problème mal posé. Cela apparaît bien sur la figure 4.

B. UN SYSTÈME D'INFÉRENCE PROBABILISTE

Nous avons programmé un système générique d'inférence probabiliste, que nous avons appliqué à l'expérience de la bassine lumineuse.

B.1. Définition et but

Un système d'inférence $\{\vartheta,C\}$ est un ensemble ϑ de variables $V_1 \ldots V_n$ discrètes et une connaissance C permettant de calculer des DDP de la forme $p_c(A|B=b,D)$, où A et B sont des sous-ensembles disjoints de variables de ϑ, b un ensemble de valeurs particulières assignées aux variables de B, et D les informations expérimentales disponibles, calculées à partir de valeurs expérimentale de certaines variables de ϑ.

En pratique, on demande souvent de ne calculer que certaines fonctions de cette DDP, typiquement :

- $p_c(A=a|B=b,D)$ pour une valeur particulière a des variables de A,
- la valeur a_o la plus probable de $p_c(A|B=b,D)$,
- l'espérance <A> de cette même DDP,
- une valeur tirée aléatoirement selon la loi déterminée par cette DDP.

Par exemple le système de la bassine lumineuse comporte trois variables, ϑ = {`angle`, `cell`, `env`}, et dispose des données expérimentales `data`. On veut se poser des questions comme :

- Quelle est la valeur la plus probable de p_c (`angle` | `cell`=768, `env`=ξ_1, `data`) ? On cherche ainsi la commande à exécuter pour avoir le plus de chance de lire 768 sur la cellule.
- Quelle est la distribution p_c (`env` | `cell`=768, `angle`=15°, `data`) ? On cherche ainsi à savoir les chances de chaque l'environnement au vu d'une mesure du couple (`angle`, `cell`).
- Quelle est l'espérance de p_c (`cell` | `env`=ξ_2, `data`) ? Autrement dit, l'environnement ξ_2 est-t-il très illuminé en moyenne ?

B.2. La connaissance C

Pour programmer un système d'inférence tel que ci-dessus, deux questions se posent :

- Comment décrire la connaissance C permettant de calculer ces inférences (notons que dans C doit se trouver entre autres la façon dont les données expérimentales sont prises en compte) ?
- Comment décrire et obtenir les données expérimentales `data` ?

Nous allons décrire formellement C comme composée de :

- Une *structure de dépendances,* qui décrit les relations de dépendance entre les variables,
- Un ensemble de *formes paramétriques,* qui décrivent la forme de ces relations,
- Un *mécanisme d'apprentissage,* qui décrit comment les identifier expérimentalement.

Structure de dépendances

> *Définition* : pour un choix de variables $V_1 \ldots V_n$ et un ensemble D de données expérimentales, nous appelons “structure de dépendances” toute décomposition de la DDP conjointe $p_c(V_1 \ldots V_n|D)$ en un produit de DDP élémentaires connues ayant toutes D en partie droite. D étant toujours présente, nous adopterons la notation allégée $P_c(\ldots|\ldots)$ pour $p_c(\ldots|\ldots D)$.

Par exemple, pour la bassine lumineuse, la structure de dépendance est :

[7] $P_c(\texttt{angle cell env}) = P_c(\texttt{angle})\, P_c(\texttt{env})\, P_c(\texttt{cell} \mid \texttt{angle env})$.

Une telle décomposition, puisqu'elle suffit à déterminer la DDP conjointe, suffit a-fortiori à calculer toutes les inférences imaginables sur les variables considérées.

Notons que cette formule [7] implique toutes les hypothèses d'indépendances telle que $P_c(\texttt{angle} \mid \texttt{env}) = P_c(\texttt{angle})$. En effet :

$$P_c(\texttt{angle} \mid \texttt{env})$$
$$= \frac{P_c(\texttt{angle env})}{P_c(\texttt{env})} = \frac{\sum_{\texttt{cell}} P_c(\texttt{angle env cell})}{P_c(\texttt{env})} = P_c(\texttt{angle}) \sum_{\texttt{cell}} P_c(\texttt{cell} \mid \texttt{angle env})$$
$$= P_c(\texttt{angle})$$

Plus généralement, on peut voir la décomposition d'une DDP conjointe comme essentiellement une façon d'exprimer des hypothèses d'indépendance conditionnelle entre variables (d'où le terme “structure de dépendances” que nous avons adopté pour cette décomposition). La structure de dépendances concentre donc une grande partie du caractère explicatif du système.

Formes paramétriques

Une structure de dépendances est une décomposition en DDP *connues.* Il faut donner la forme fonctionnelle (paramétrique) de chacune des ces DDP. Par exemple, pour la structure [7] il faut préciser comment on calcule $p_c(\texttt{angle})$, $p_c(\texttt{env})$ et $p_c(\texttt{cell} \mid \texttt{angle env data})$. Reprenant notre caractérisation [6] de l'expérience de la bassine, nous avons :

$$\begin{cases} p_c(\texttt{angle}) = 1/71 \\ p_c(\texttt{env}) = 1/3 \\ p_c(\texttt{cell} \mid \texttt{angle env data}) = G\,(\texttt{cell} \mid \mu(\texttt{angle,env,data}), \sigma(\texttt{angle,env,data})). \end{cases}$$

Certains des paramètres qui apparaissent sont fixes (1/71, 1/3), mais d'autres sont variables en fonction des données (μ et σ).

C'est dans la détermination des formes paramétriques qu'intervient le principe de MaxEnt. En terme de MaxEnt, les paramètres de nos fonctions correspondent aux observables qui ont permis de déterminer leur forme. Ici, l'apparition des paramètres μ et σ dans la troisième forme correspond au choix des observables `cell` et $\texttt{cell}^2$ dont nous avons déduit la forme gaussienne (et un paramètre constants comme 1/71 correspond au choix du simple nombre de valeurs possibles pour la variable, qui mène au principe d'indifférence).

Mécanisme d'apprentissage

Les données expérimentales servent à calculer les paramètres, dans notre exemple μ(`env`, `angle`, `data`) et σ(`env`, `angle`, `data`). Pour réellement parler d'apprentissage, il faut rappelons-le que ces calculs se traduisent comme la mise à jour des paramètres en fonction de nouvelles données, et donc aient une forme itérative : si de nouvelles données sont disponibles et que `data` se transforme en $\texttt{data}^+$, alors μ(`env`, `angle`, $\texttt{data}^+$) doit se calculer à partir de μ(`env`, `angle`, `data`) et des nouvelles données, et de même pour σ.

La spécification de ces calculs de μ et σ est un “mécanisme d'apprentissage”.

Dans notre exemple, ces paramètres sont identifiés comme les moyennes et écarts-types expérimentaux des mesures de cell. Pour déterminer ces valeurs itérativement, nous procédons ainsi : pour chaque valeur de (`angle`, `env`) nous gardons en mémoire :

- le nombre n de données ayant servi à calculer μ et σ jusqu'à présent,
- la somme $S_1=\sum \mathit{cell}$ des valeur de `cell` lues jusqu'à présent,
- la somme $S_2=\sum \mathit{cell}^2$ des carrés de ces valeurs.

Ces valeurs sont mises à jour facilement à chaque nouvelle mesure, et les paramètres μ et σ en sont déduits instantanément : $\mu = S_1/n$, et $\sigma^2 = (1/n)(S_2 - S_1^2/n)$.

Le lien entre données et paramètres

Comment à présent décrire et obtenir les données expérimentales `data` destinées à identifier notre système ?

Une première idée serait de considérer ces données comme l'ensemble de toutes les mesures de `angle`, `env` et `cell` que l'on a faites. Mais cette solution est peu économique : on voit bien dans l'exemple ci-dessus que pour effectuer les calculs, il suffit de ne retenir que le minimum d'informations qui permettent de calculer itérativement les observables. Il suffit donc de prendre pour `data` l'ensemble des triplets (n, S_1, S_2) associés à chaque valeur de (`angle`, `env`).

En pratique, c'est donc au niveau du mécanisme d'apprentissage qu'est déterminée la représentation des informations expérimentales et la façon de les mettre à jour itérativement.

B.3. Les mécanismes d'inférence

Notre système d'inférence $\{\vartheta, C\}$ étant spécifié, il faut l'utiliser pour faire des inférences. Disposant des manipulations probabilistes élémentaires du chapitre 3 (règles du produit et de la somme, formule de Bayes, normalisation), nous cherchons une méthode automatique pour transformer une question du type $P_c(\ldots|\ldots)$ en une chaîne de calculs.

Principe général

Dans une question, une variable donnée peut être soit en partie gauche, soit en partie droite, soit absente. Nous appellerons respectivement X, Y et Z les trois sous-ensembles ainsi définis (qui forment une partition de ϑ), et x, y et z sont des ensemble de valeurs particulières de ces variables. La question générique est : trouver une fonction de $P_c(X|y)$, par exemple sa valeur en un point, sa moyenne, ou sa valeur la plus probable.

La méthode brutale est la suivante. D'une part, nous nous posons toujours des questions du type $P_c(x|y)$, c'est-à-dire la probabilité d'une valeur précise de X, et pour calculer une valeur la plus probable ou une moyenne, nous procédons itérativement (rappelons que nos variables sont toutes discrètes). Alors, puisque nous savons calculer la DDP conjointe $P_c(XYZ)$, on a :

[8] $$P_c(x|y) = \frac{\sum_Z P_c(xyZ)}{\sum_{XZ} P_c(XyZ)}.$$

Notons que si nous ne cherchons que la valeur la plus probable de cette DDP, la constante de normalisation n'intervient pas dans le sens des comparaisons, et le dénominateur n'a pas à être calculé. De même, si nous cherchons une valeur moyenne, le dénominateur peut n'être calculé qu'une seule fois pour toutes les valeurs x de X.

Une telle méthode est purement calculatoire et ne tient pas compte des simplifications générique que peut apporter une décomposition donnée. Nous avons alors trois options :

- Écrire un système qui effectue d'abord les transformations symboliques, trouve les simplifications à faire, et ensuite seulement passe aux calculs numériques ;
- Programmer, pour chaque structure de dépendance, des mécanismes spécifiques d'inférence ;
- Abandonner l'aspect “universel” de notre caractérisation des systèmes d'inférence probabiliste, et chercher un type générique de structure de dépendance qui soit manipulable efficacement tout en restant assez riche pour garder une utilité générale.

Notre système actuel se contente de la seconde solution (mécanismes spécifiques). La première (calcul symbolique) est en cours de réalisation dans notre équipe (Pierre Bessière, non encore publié). Un exemple de la troisième solution, les réseaux bayésiens, sera évoqué plus loin dans ce chapitre.

Pour notre part, nous avons délibérément choisi de coder des mécanismes spécifiques pour chaque structure de dépendance, pour les raisons suivantes :

- Nous n'utilisons actuellement que des systèmes très simples comportant deux ou trois variables, et il n'y a qu'un nombre limité de structures de dépendances que nous utilisions. Ce n'est donc pas gênant en pratique d'avoir à coder des mécanismes spécifiques.
- Cette solution est extrêmement naturelle à implémenter avec un langage objet. La structure du logiciel permet de programmer vite et bien tout nouveau mécanisme d'inférence. De plus, les méthodes spécifiques restent assez proches de la méthode générale, pour qu'il s'agisse de “variantes” plutôt que d'une reconception complète, et l'adaptation est assez facile à faire. Le système de base et les variantes spécifiques que nous utilisons pour nos expériences tiennent dans moins de 40K de source Lisp commentée.
- À l'avenir, nous ne pouvons pas encore dire si nous aurons à utiliser des systèmes d'inférence complets utilisant de nombreuses variables, ou si nous chercherons plutôt à utiliser de petits systèmes comme des “boîtes à outils” dont l'interaction fera appel à des formalismes d'autre nature (problèmes de décision, mécanismes incrémentaux ou sélectionnistes, systèmes dynamiques). Il est trop tôt pour se prononcer, et donc pour se lancer dans la réalisation d'un logiciel très général ou très complexe. Pour le moment, notre système actuel semble largement suffisant et peut encore subir de nombreux ajouts avant de se révéler lourd à utiliser et nécessiter une solution de rechange.

Exemples

Montrons sur les exemples de questions posées au §B.1 comment marche ce système :

- Quelle est la valeur la plus probable de P_c (`angle` | `cell`=768, `env`=ξ_1) ?

La formule [8] nous donne :

P_c (*angle* | *cell env*) = P_c (*angle cell env*) / $\sum_{\text{angle}} P_c$ (angle *cell env*).

Puisque l'on cherche une valeur la plus probable, le dénominateur est inutile. Il s'agit donc de trouver la valeur $angle_0$ maximisant P_c(`angle` *cell env*) sur l'ensemble des valeurs de `angle`. Cette expression se décompose en P_c (`angle`) P_c (*env*) P_c (*cell* | `angle` *env*). C'est là que la structure de dépendance spécifique [7] intervient pour simplifier cette expression, éliminant P_c(*env*) qui est une constante par rapport à `angle`. L'élimination des a-prioris constants a aussi été prévue étant donné la fréquence de ce cas, et ici P_c(`angle`) disparaît aussi, ne laissant que le terme P_c (*cell* | `angle` *env*), qu'on maximise alors sur `angle` (notons que ce n'est une gaussienne qu'en fonction de *cell*, pas de `angle`).

- Quelle est la distribution P_c (`env` | `cell`=768, `angle`=15°) ?

Pour calculer cette DDP nous allons simplement chercher les probabilités pour les trois valeurs ξ_1, ξ_2 et ξ_3 que peut prendre `env`. On a d'après la formule [8] :

P_c (env | `cell angle`) = P_c (env `cell angle`) / $\sum_{env} P_c$ (env `cell angle`).

Ici le dénominateur est nécessaire, mais n'est calculé qu'une seule fois, ainsi que les termes constants dans la formule de décomposition [7].

- Quelle est l'espérance de P_c (`cell` | `env`=ξ_2) ?

La formule [8] donne cette fois :

P_c (*`cell`* | *env*) = $\sum_{angle} P_c$ (`angle` *`cell`* *env*) / $\sum_{angle,cell} P_c$ (`angle cell` *env*),

et l'espérance est :

E (`cell` | *env*) = $\sum_{cell}$ `cell` P_c(`cell` | *env*).

Après décomposition, les termes constants sont éliminés automatiquement entre numérateur et dénominateur (ceci est un mécanisme spécifique), et le dénominateur n'est calculé qu'une seule fois.

Une amélioration : les conditions booléeenes

Le système d'inférence ainsi réalisé peut très facilement être étendu pour répondre à des questions de la forme P_c(X|YK), X et Y étant des ensembles disjoints de variables, et K étant une fonction booléenne des variables de ϑ. L'ajout de cette condition transforme juste la formule [8] en :

[9] $$P_c(x|yK) = \frac{\sum_Z P_c(xyZ)\ K(xyZ)}{\sum_{XZ} P_c(XyZ)\ K(XyZ)}$$ (K étant une fonction qui retourne 0 ou 1).

Par exemple, on peut ainsi demander P_c (`cell` | `angle`<0°, `env`=ξ_1).

Le système peut refuser certaines conditions, notamment celles portant sur `cell`, qui demanderaient des calculs complexes (intégration de gaussiennes entre des bornes autres qu'infinies) que nous n'avons pas voulu programmer pour garder cette extension très simple.

B.4. Implémentation informatique

L'implémentation est faite en CommonLisp orienté objet (CLOS) ; et la structure de classes est la suivante :

Un *système d'inférence* est défini comme tout objet acceptant une méthode proba, qui permet de poser génériquement des questions du type $P_{...}$(...|...). Ne manipulant le plus souvent seulement une, deux ou trois variables, des méthodes spécifiques proba1, proba2 et proba3 sont généralement utilisées au lieu de la méthode générale proba.

Un *système de dépendances* est défini comme tout système d'inférence pour lequel ont été définies des mécanismes spécifiques pour effectuer les inférences. Les systèmes de dépendances acceptent génériquement les méthodes ddp, proba-max, espérance et aléa, qui retournent respectivement une DDP complète, la valeur la plus probable d'une DDP, l'espérance d'une DDP et une valeur tirée aléatoirement selon une DDP. Une méthode utile est aussi pseudo-proba, qui retourne une valeur non normalisée : lorsqu'on veut mettre en rapport deux valeurs et que les constantes de normalisation s'éliminent dans ce rapport, cela évite des calculs inutiles.

Ensuite, un *système d'apprentissage* est défini comme tout système de dépendances ayant une forme paramétrique déterminée et un mécanisme d'itentification des paramètres. Un système d'apprentissage est manipulé par les méthodes `reset-apprentissage`, `apprendre` et `apprendre-offline`, qui respectivement initialise le système dans le cas de zéro données, met à jour les paramètres au vu d'une nouvelle donnée, et identifie globalement le système sur un ensemble de données pré-enregistrées.

Indépendemment de cette structure hiérarchique, nous avons par ailleurs défini des *codages,* qui essentiellement discrétisent des entrées réelles, via une méthode `coder`, ou au contraire décodent un indice discret en une valeur réelle, via une méthode `décoder`. Enfin, des *prétraitements* permettent de transformer des données sensorielles en données pouvant nourrir un système d'apprentissage. Les prétraitements s'utilisent via les méthodes `observable`, qui de données expérimentales brutes ne retient que les observables nécessaires aux calculs, et `observables-offline`, qui détermine `data` d'après toutes les données brutes pré-enregistrées.

Il existe de plus à chaque fois des méthodes de confort et de mise au point : visualisations diverses, vérification des normalisations, etc…

C. RÉSULTATS : INCERTITUDES ET CONDITIONS DE VALIDITÉ

L'expérience de la bassine lumineuse est conceptuellement extrèmement simple, mais dans la pratique elle a donné quelques résultats étonnants, dûs au fait qu'il s'agit d'un environnement physique (ces résultats n'auraient jamais pu être observés dans une simulation conçue de façon habituelle, voir chapitre 2 §J.1). L'élément inattendu est l'importance considérable du protocole d'exploration de la bassine pour recueillir les données de l'apprentissage. C'est une manifestation du problème de conditions de validité, décrite au §C.1. Cela nous amènera à nous interroger sur la nature de la modélisation effectuée, et à souligner la distinction essentielle entre les notions d'hypothèse et de connaissance, qui a des conséquences importantes sur notre approche (§C.2).

C.1. Les caprices d'une cellule photo-électrique

Nous avons “triché” à un égard dans notre description de l'expérience. Nous avons prétendu que le protocole d'exploration était réalisé aléatoirement, et que l'a-priori p_c(`angle`) équiprobable reflétait ce choix. Mais en fait, pour gagner du temps, l'exploration était faite de façon systématique en parcourant les angles de -175° à +175° dans l'ordre. Le choix d'un a-priori uniforme ne modélisait donc en fait que notre volonté de ne pas prendre en compte cette connaissance.

La figure 5 montre les résultats que donne l'expérience de la bassine lumineuse avec un protocole d'exploration effectivement aléatoire.

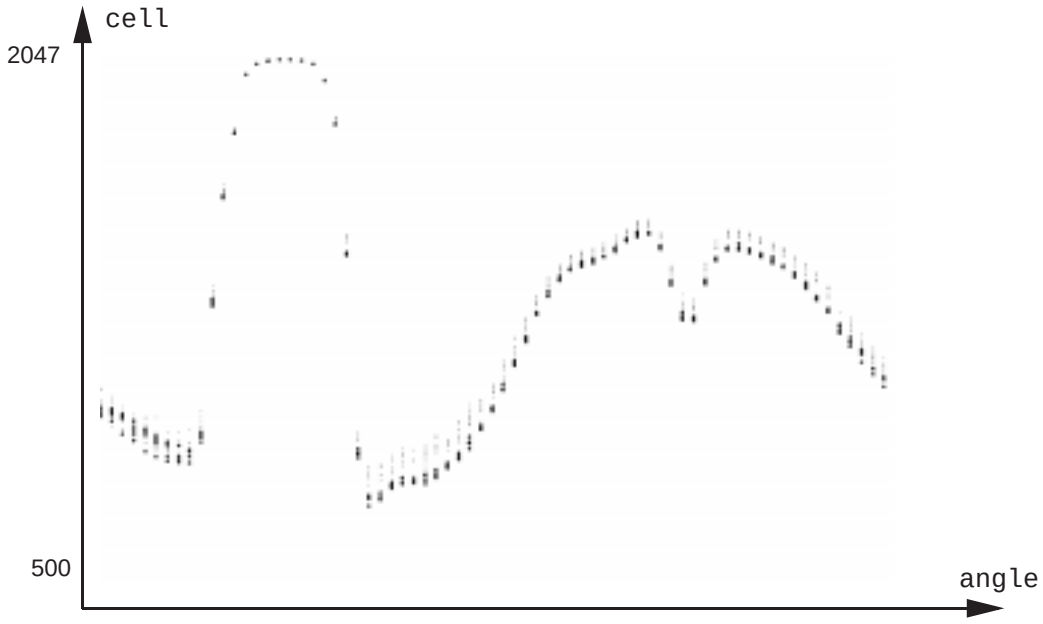


Figure 5. Mesures enregistrées avec un protocole aléatoire (points fins), superposées aux mesures de la figure 2a (points gras) qui provenaient d'un balayage systématique des angles. La seule différence entre ces mesures est dans le protocole utilisé. En particulier, les mouvements aléatoires amènent régulièrement la cellule face à la lampe, ce qui lui maintient une excitation globale supérieure au cas d'un protocole systématique.

Devant ce phénomène, nous avons découvert que les cellules utilisées sont sujettes à des phénomènes d'hystérésis très lents à amortir. Ce sont des résistances variables sensible à la température, et la proximité d'une lampe halogène de 6 watts est sans doute en grande partie responsable de ce fonctionnement étrange.

Dans ces conditions, adopter un protocole d'exploration systématique plutôt qu'aléatoire réduit énormément les variations constatées. L'historique des positions successives de la cellule étant à peu près constante, les variations dûes à l'hystérésis sont largement atténuées.

Le phénomène nous avait à l'origine échappé, pour la raison suivante.

Pour évaluer nos résultats nous avons d'abord utilisé la “validation croisée”, presque universellement adoptée dans les travaux sur l'apprentissage. Toutes les données étant enregistrées, nous les partitionnons en deux ensembles “d'apprentissage” et “de test” (typiquement un partage moitié-moitié). L'ensemble d'apprentissage sert à identifier les paramètres, et l'ensemble de test à vérifier les performances sur des données “nouvelles”.

L'erreur quadratique moyenne obtenue pour les prédictions de `cell` à partir de `angle` varie entre 8.6 et 11.1 selon les partitionnements apprentissage-test, sur des valeurs allant de 700 à 2047. Ces résultats semblent donc très satisfaisants.

Cependant, par acquit de conscience, nous avons fait des mesures avec l'expérience suivante : on tire aléatoirement une valeur désirée pour `cell`, on infère une valeur de `angle`, on se rend physiquement à cette position pour lire la valeur effective, on calcule l'erreur obtenue par rapport à la valeur souhaitée, et on moyenne ces erreurs sur un grand nombre d'essais.

L'erreur obtenue est de 71.5. Après enquête sur ces mauvais résultats, nous avons découvert le fonctionnement inattendu de la cellule.[3]

En utilisant pour l'apprentissage les données de la figure 5, donc obtenues avec un protocole aléatoire, l'erreur devient de 44.9. Cela reste moyen, mais plus normal par rapport aux écarts-types appris par la DEP (38.3 en moyenne, contre 4.9 avec le protocole systématique).

Nous avons tiré deux leçons de cette expérience :

- Les seules validations correctes d'un apprentissage en robotique sont celles conduites *expérimentalement et on-line,* non celles calculées sur un ensemble déjà enregistré. Dans un environnement non systématiquement contrôlable, il y a toujours un risque que le protocole d'apprentissage se révèle inadéquat lors de l'utilisation attendue du modèle, des phénomènes parasites imprévus étant toujours possible. Une validation réaliste est celle qui donne à ces phénomènes l'occasion de se manifester, et donc au concepteur l'occasion d'en tenir compte par la suite.
- Nous avions établi notre modèle en pensant que, vu le dispositif expérimental, une position donnée de la cellule donnerait toujours à peu prés la même lecture de la cellule. Cela se révèle moins vrai que prévu, mais le modèle demeure utilisable : il permet encore de faire des prédictions, et elles ne sont pas trompeuses puisque explicitement associées à des incertitudes élevées. Un modèle où cette gestion de l'incertitude serait absente, comme un réseau de neurones ou un modèle tiré des lois de la physique, serait plus trompeur : ses prédictions seraient tout aussi approximatives, mais rien n'aurait été prévu pour moduler leur exploitation en fonction de ce manque de fiabilité.

C.2. Le choix d'un modèle : hypothèses ou connaissances ?

Si nous représentons les données brutes sous la forme d'histogrammes, elles n'ont pas toujours une allure gaussienne (c'est peu fréquent mais cela est arrivé). Par exemple, lors d'une expérimentation antérieure réalisée avec un protocole systématique, nous avions trouvé les histogrammes systématiquement bimodaux de la figure 6.

---

[3] Le problème était à l'origine bien plus impressionnant, les écarts-types moyens de la DEP obtenue par un protocole aléatoire étant de 263. Le protocole systématique permettait alors de réduire les incertitudes à un niveau comparable à ce que nous obtenons maintenant avec un protocole aléatoire. La chute de performance entre cross-validation et test réel était alors sidérante. Le problème a été en grande partie corrigé par le changement de la cellule, déteriorée à force de subir ces phénomènes, et par une modification du montage électronique en amont de la numérisation du signal.

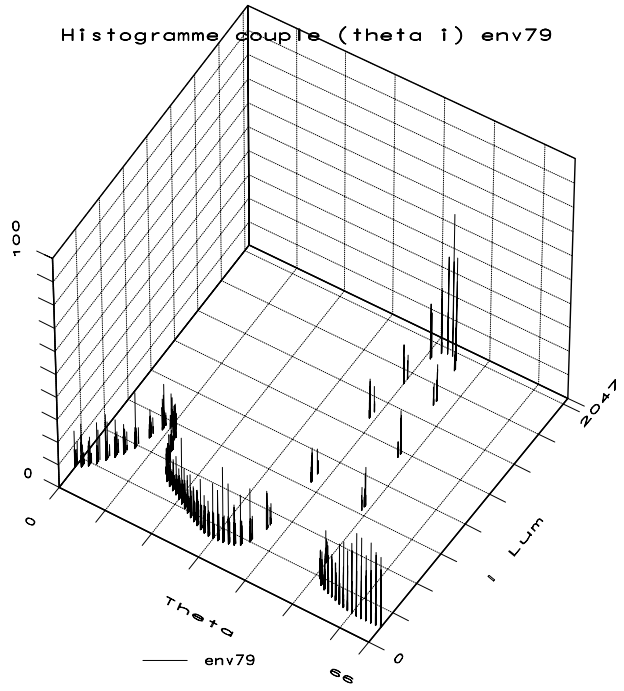


Figure 6. Le détail des histogrammes bimodaux obtenus lors d'une expérience antérieure. Sur cette figure, théta désigne `angle`, lum désigne `cell`, et sur l'axe vertical on lit le nombre d'occurence d'une certaine mesure de `cell` pour un `angle` donné.

Voyant cela, nous avons trouvé plusieurs explications plausibles :

- la bassine avait un peu bougé pendant les mesures, et tous les pics s'étaient légèrement décalés. Le résultat final était la superposition des deux séries de mesures, avant et après décalage.
- certaines lumières environnantes (néons) avaient été éteintes ou allumées au cours de l'expérience par une des nombreuses personnes travaillant dans la salle robotique où se déroulait l'expérience de façon automatique.

Les histogrammes n'avaient rien de gaussiens — et pourtant nous avons maintenu notre modèle tel quel. En effet, les autres possibilités étaient d'une part de maintenir le modèle mais de prendre d'autres données qui y soient adaptées, d'autre part de modifier le modèle pour prendre en compte le phénomène parasite. Or dans tous les cas cela pose des problèmes de fond.

Fallait-il, comme on nous l'a conseillé une fois, éliminer de nos données les mesures “parasites” avant d'identifier le modèle ? La perturbation n'ayant pas été contrôlée, nous ne savons pas dire quelle est la situation censée être “normale” et quelles mesures il faut garder. De plus, rien ne nous dit qu'une perturbation similaire ne pourra pas se reproduire puisqu'elle s'est déjà produite, et ignorer cette éventualité en éliminant les données où elle se manifeste relève de la “politique de l'autruche”.

Fallait-il, comme on nous l'a souvent dit, oublier toutes les mesures ainsi “faussées” et reprendre l'expérience en faisant plus attention aux conditions expérimentales ? Cela signifie que si l'environnement n'a pas été assez bien contrôlé

l'information obtenue est non pertinente et doit être ignorée. Cette position est incompatible avec notre objectif d'automatiser le fonctionnement d'un robot en environnement non contrôlé, puisqu'elle consiste à éluder le problème qui nous intéresse. Notons de plus que le modèle obtenu s'est révélé tout à fait exploitable.

La solution restante est de modéliser le phénomène parasite pour en tenir compte à l'avenir. Ceci introduit des paramètres supplémentaires, qu'il faut donc identifier expérimentalement, et il faut donc dire comment faire cette identification. Par exemple, si on choisit pour modèle la superposition de deux gaussiennes, cela fait pour chaque couple (*angle, env*) quatre paramètres au lieu de deux (un μ et un σ par gaussienne). Cependant, vu les explications du phénomènes proposées plus haut, rien n'exclut la possibilité de données expérimentales trimodales, et peut être faut-il prévoir cette éventualité… Quel modèle précis faut-il choisir, en fin de compte ?

Cette question est difficile. Or, si une révision du modèle est possible, elle n'est pas *impérative* comme on nous l'affirme souvent. Il est en effet possible de prendre en compte ce genre de phénomène par la notion d'incertitude, et de garder notre premier modèle gaussien.

La clef de cette démarche, c'est que ce modèle gaussien traduit non pas une *hypothèse,* mais une *connaissance.* Nous ne postulons pas que la distribution observée expérimentalement aura une forme gaussienne, mais nous constatons plutôt que cette forme gaussienne est le meilleur moyen de représenter formellement notre connaissance. Il ne s'agit donc pas d'une hypothèse *à vérifier* sur la nature d'un processus physique, mais de la *spécification* de la façon dont fonctionne notre robot — de même qu'un programme informatique ne traduit pas des hypothèses faites par le programmeur, mais ses connaissances. Cette distinction entre hypothèses et connaissances est fondamentale pour notre démarche.

Pour donner un exemple, revenons au §A.2 (description du modèle probabiliste). Nous avions justifié le choix d'une distribution gaussienne pour p_c(`cell` | `angle env data`) par le principe de MaxEnt, et admis temporairement que cette façon de présenter les choses pouvait sembler artificielle et inutilement compliquée, tant l'hypothèse gaussienne semble s'imposer dans un tel problème. La raison pour laquelle nous tenons à cette démarche est qu'elle traduit justement cette distinction entre hypothèses et connaissances. Le principe de MaxEnt dit que le type de connaissance que nous avons se formalise par une gaussienne ; *tant que nous n'avons pas acquis une connaissance qui permette de prédire une autre forme, la gaussienne reste la meilleure formalisation que nous puissions considérer.* Cette formalisation est indépendante des données ; celles-ci ne peuvent la remettre en question, elles ne peuvent qu'*amener le concepteur* à remettre en question la connaissance qu'il a voulu formaliser. Ce sont ces considérations qui justifient notre référence constante au principe de MaxEnt, bien qu'elle soit inutile d'un point de vue technique.[4]

Quand au fait que le modèle se révèle non satisfaisant par rapport à ce que nous pensions, il faut noter que *si quelques observations empiriques peuvent suffire à*

---

[4] Cette optique justifie aussi une phrase qui avait pu choquer certains lecteurs : nous avons au paragraphe précédent admis que malgré l'emploi d'un protocole d'exploration systématique, nous avions gardé l'a-priori p_c(`angle`)=1/71, et que cet a-priori ne représentait donc pas le protocole utilisé mais seulement *notre volonté de ne pas prendre en compte la connaissance de ce protocole.* En termes d'hypothèse, cet a-priori pouvait choquer puisque le protocole était tout à fait déterministe. En termes de connaissance, nous avons fait un *choix* de modélisation, qui se traduit simplement par un a-priori équiprobable.

faire douter d'un modèle, pour décider de ne pas l'utiliser il faut soit décider de ne pas agir du tout, soit avoir un modèle de rechange (opérationnel, i.e. pas seulement une explication informelle des raisons de la médiocrité du premier).[5] En robotique autonome, nous serons rarement dans un des ces deux cas, et il est important d'exploiter les modèles dont on dispose, même imparfaits.

La prise en compte de l'incertitude expérimentale est un recours permettant de continuer à utiliser au mieux un modèle imparfait. Ce n'est certes pas une panacée, et la question de la détermination de modèles de rechange reste essentielle. Mais la révision d'un modèle ne peut être que le fait du concepteur. Cela ne concerne donc plus le cycle d'adaptation. Au chapitre suivant, nous proposerons d'aborder ce problème de la révision du modèle par le cycle de l'incrémentalité, de façon à tirer partie de l'imprévu pour enrichir les possibilités du robot.

C.4. Le cas des environnements inconnus

Que se passe-t-il lorsque l'environnement devient un environnement inconnu ξ ? Dans le modèle utilisé jusqu'à présent, cette éventualité n'est même pas reconnue. Or, il est clair qu'en utilisant ce modèle on peut se rendre compte, empiriquement, que les prédictions sont mises en défaut. Ce paragraphe décrit comment nous avons modifié notre modèle pour prendre en compte le cas d'un environnement inconnu.

Il faut commencer par reconnaître *la possibilité même* de tomber sur des environnements inconnus. Pour cela, prenons comme ensemble de valeurs de `env` l'ensemble $\{\xi_i, i=1\ldots N\}$. N est une borne supérieure du nombre total d'environnements possibles ; s'il n'y a pas de bornes alors $N=\infty$. Nous ne connaissons que 3 de ces N possibilités : ξ_1, ξ_2 et ξ_3.

Il devient évidemment nécessaire de redéfinir l'a-priori $p_c(\mathrm{env})=1/3$. Accordons par exemple 10% de chances pour un environnement de n'être aucun des trois connus, et prenons comme DDP a-priori 30% pour chacun de ces derniers, et selon le principe d'indifférence 10/(N-3)% pour les autres possibilités.[6]

Dans ce nouveau cadre, les paramètres μ et σ ne sont définis que pour ξ_1, ξ_2 et ξ_3, mais sont indéfinis pour toute autre valeur. Imaginons alors qu'après avoir mesuré des données (*angle, cell*) nous cherchions à savoir si l'environnement est ξ_1, ξ_2, ξ_3, ou bien un autre encore inconnu. Nous pouvons calculer $p_c(\xi_1 \mid angle\ cell\ \mathrm{data})$, $p_c(\xi_2 \mid angle\ cell\ \mathrm{data})$, $p_c(\xi_3 \mid angle\ cell\ \mathrm{data})$, mais nous sommes incapables de calculer p_c (“$\mathrm{env}\neq\xi_1$ et $\mathrm{env}\neq\xi_2$ et $\mathrm{env}\neq\xi_3$” $\mid angle\ cell\ \mathrm{data}$), car cette probabilité se calcule comme $\sum_i p_c(\xi_i \mid angle\ cell\ \mathrm{data})$ pour i=4…N, et que tous ces termes restent indéfinis.

---

[5] Un adage commun est que “les données parlent pour elles-mêmes”. Selon ce point de vue, la forme des données pourrait se substituer à l'acte de modélisation, au *choix* du concepteur. Il est clair que dans ce cas, la meilleure distribution est celle qui coïncide exactement avec l'histogramme des valeurs expérimentales (leurs fréquences individuelles). Par exemple, le fait de décider d'une distribution gaussienne *parce que* les données ont une ”allure en cloche” est vide de sens. On décide d'une forme gaussienne en raison du maximum d'entropie ou de la loi des grands nombres, pas en fonction des données. Les données permettent de juger automatiquement d'un modèle, de mesurer automatiquement son efficacité, mais pas de le déterminer automatiquement : cela ne peut être fait actuellement que par l'intelligence humaine.

[6] Le cas “$N=\infty$” signifie alors que N est pris arbitrairement grand mais fini, et qu'il sera éliminé des résultats finaux par un passage à la limite si les normalisations intermédiaires ne s'en sont pas chargées.

Ce qui se passe, c'est que l'hypothèse "l'environnement est inconnu" ne permet en soi aucune prédiction. On ne peut donc pas la comparer à une autre, puisqu'aucune de ses conséquences n'est calculable.

L'idée couramment adoptée est alors de poser un seuil arbitraire pour y comparer la probabilité de "`env`=ξ_1 ou `env`=ξ_2 ou `env`=ξ_3", qui est calculable, et de décider, si cette probabilité est inférieure au seuil, que l'environnement doit être considéré comme inconnu. Cette heuristique oblige à faire appel à des notions de décision, donc de sortir de la position théorique consistant à modéliser un état de connaissance. Or, nous pouvons trouver une autre heuristique qui s'exprime dans notre cadre théorique.

Introduisons à cet effet un environnement supplémentaire, noté ξ_0, qui est un environnement "blanc" dans lequel toutes les valeurs de `cell` sont équiprobables indépendemment de celles de `angle`. Cet environnement peut alors être comparé aux autres, et par exemple devenir le plus probable, ce qui signifie alors que les données observées ont plus de chance d'avoir été tirées au hasard que d'avoir été produites par n'importe lequel des environnements connus. Nous décidons alors que dans ce cas nous sommes dans un environnement inconnu.

Par cette méthode, c'est au niveau de l'interprétation du modèle que l'on introduit des éléments supplémentaires, non au niveau du modèle formel lui-même. L'aspect "décision" reste implicite et informel, et nous restons dans le pur cadre de la représentation probabiliste d'un état de connaissance. Le problème du choix arbitraire d'un seuil est remplacé par celui du choix a-priori de p_c("env=ξ_0").

D. INFÉRENCE PAR RÉSEAUX BAYÉSIENS

La caractérisation des systèmes d'inférences présentée à la section B suit de près les notions théoriques. Elle est concise et facile à utiliser. Cependant, nous avons vu que pour gérer un grand nombre de variables elle ne sera pas forcément très bien adaptée. Nous avons alors cité deux idées pour réaliser un système adapté à des utilisations plus complexes : effectuer les calculs sous forme symbolique, ou restreindre la généralité du système pour privilégier son efficacité. Les réseaux bayésiens que nous présentons à présent sont un exemple de cette seconde approche, un compromis intéressant entre efficacité des calculs, généricité des connaissances formalisables, et richesse des inférences calculables (et donc des exploitations possibles du système une fois celui-ci réalisé).

Les réseaux bayésiens (Pearl 1991) sont une réponse au problème de l'organisation pragmatique de calculs probabilistes bayésiens complexes. Ces travaux ont été développés dans le contexte du diagnostic médical, et accordent une importance particulière à l'explicabilité des architectures obtenues.

L'organisation en réseau bayésien est fondée sur la notion qualitative d'indépendance conditionnelle. A est indépendant de B conditionnellement à C si p(A|BC)=p(A|C). Pearl affirme que cette notion est fondamentale en tant que façon naturelle d'organiser nos connaissances :

> «Conditional independance is not a grace of nature for which we must wait passively, but rather a psychological necessity which we satisfy actively by organizing our knowledge in a specific way» (p. 44)

«The most striking inadequacy of traditional theories of probability lies in the way these theories address the notion of independance. The traditional definition of independance uses equality of numerical quantities, as in p(xy)=p(x)p(y), suggesting that one must test whether the joint distribution of X and Y is equal to the product of their marginals to determine whether X and Y are independant. By contrast, people can easily and confidently detect dependancies, even though they may not be able to provide numerical estimates of probabilities. … Human behavior suggests that relevance information is inferred qualitatively from the organizational structure of human memory, not calculated from numerical values assigned to its components» (p. 79-80)

La notion d'indépendance conditionnelle est alors considérée comme le guide naturel permettant de décrire formellement une certaine connaissance intuitive. Cette notion est alors axiomatisée indépendemment du cadre probabiliste, en tant que notion véritablement qualitative.

Pearl présente alors un autre guide naturel du raisonnement humain, qui est l'appel courant à des représentations sous forme de graphe pour décrire les relations de dépendance. Pearl formalise alors le rapport entre les propriétés de certains graphes et celles de l'indépendance conditionnelle. Cette démarche nous semble exemplaire : il s'agit de cerner précisément ce que permet d'exprimer un graphe dans la démarche d'un concepteur, d'expliciter le type de problèmes auxquels cela le restreint, et de discuter des possibilités qu'il se ferme ainsi.[7]

Le résultat (très grossièrement exprimé) est que la représentation graphique la plus riche est celle de graphe orienté. La notion d'indépendance adoptée est donc finalement directionnelle (non symétrique), ce qui est présenté comme une analogie à la notion de causalité :

«Causation is not a property of nature but rather a mental construct devised fot the efficient organization of knowledge … we take the position that human obsession with causation, like many other psychological compulsions, is computationally motivated" (p. 382-383).

Le cadre des réseaux bayésiens ne permet d'exprimer qu'un certain type de connaissances, mais toute l'argumentation vise à montrer que ce type est riche et utile. De notre point de vue, on peut voir cela comme une classe particulière de structures de dépendances (celles qui peuvent être représentées formellement par un graphe acyclique orienté au sens défini par Pearl). Par exemple, le problème de la bassine lumineuse aurait pu être traité par le réseau de la figure 7.

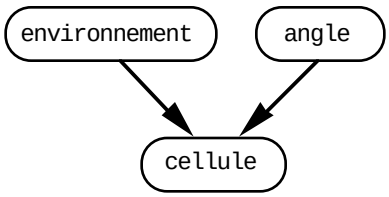


Figure 7. La bassine lumineuse sous la forme d'un réseau bayésien. Le sens de flèches représente intuitivement des relations de "causalité".

[7] Cette préoccupation est toujours présente chez Pearl : autant que possible, l'étude et la comparaison à d'autres approches des propriétés qualitatives des réseaux bayésiens s'appuye sur une formalisation axiomatique : notions d'indépendance, caractère explicatif des graphes, apports de la théorie de la décision, sémantique du raisonnement par défaut.

Pearl développe dans ce cadre des algorithmes efficaces pour mener le calcul des inférences directes et inverses dans un réseau bayésien. La complexité de calcul atteinte est linéaire en fonction de la taille du réseau, donc très satisfaisante.

Un autre intérêt de ces réseaux — et un intérêt majeur dans notre optique — est qu'ils sont conçus de façon à ce que leur structure puisse évoluer (restructuration des observables, ajout de nouvelles variables). Quelques clefs sont données pour systématiser cette évolution, mais cet aspect reste malgré tout embryonnaire. Dans la perspective incrémentale de cette thèse, c'est une ouverture notable dans une direction totalement passée sous silence dans la théorie PaL, qui ne traitait que de connaissances dont la structure est statique. La modification dynamique de la structure d'un système probabiliste est un des points délicats et essentiels qu'il nous faudra nécessairement aborder plus tard.

Pour en revenir à notre approche actuelle, la différence d'optique avec les réseaux bayésiens est qu'elle permet de représenter des types de connaissance plus divers, au détriment d'une moins grande explicabilité des processus menant aux résultats. Les applications des réseaux bayésiens ont trait essentiellement au diagnostic médical, où le raisonnement suivi doit pouvoir être exposé pour justifier les résultats. En robotique, cette contrainte de “transparence” est moins cruciale.

E. L'EXPÉRIENCE DU KHÉPÉRA TÉLÉGUIDÉ

L'approche décrite dans ce chapitre a été utilisée pour commander un petit robot mobile, nommé Khépéra (figure 8, voir Mondada & al. 1993). Ce robot est commandé par la vitesse de ses roues gauche et droite. Il dispose de huit capteurs infrarouges, 6 devant et 2 derrière, qui peuvent être lus à la fois de façon passive (capteurs de lumière) et active (proximètres). Des capteurs proprioceptifs permettent de plus de déterminer le mouvement effectif des roues.

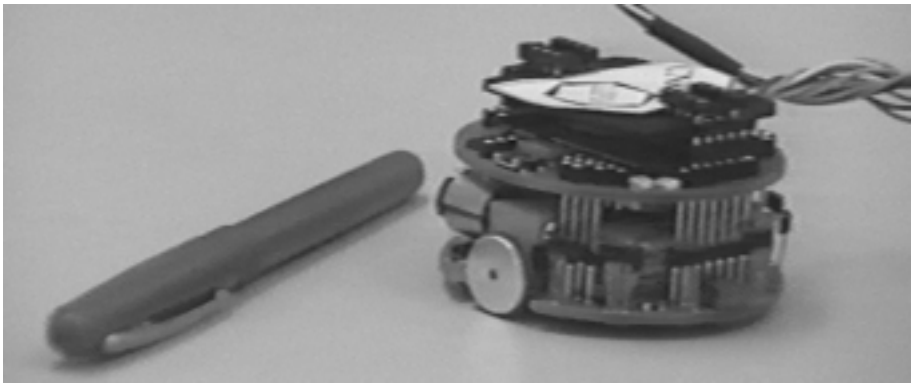


Figure 8. Le robot Khépéra (diamètre 57 mm).

L'idée de l'expérience est de télécommander le Khépéra, grâce à un joystick, en lui faisant suivre des trajectoires sans collisions dans un environnement encombré. Les données expérimentales ont été obtenues de quatre opérateurs différents. Elles ont permis d'apprendre une relation sensori-motrice caractérisant l'interaction du robot avec son environnement lors de ce comportement télécommandé. Nous avons ensuite utilisé cette relation pour programmer un comportement automatique d'évitement d'obstacles.

Les variables considérées pour l'apprentissage sont les suivantes :

- *Commande du robot.* Khépéra est commandé par les vitesses indépendantes de ses roues gauche et droite. Appelons “vitesse de translation” la somme des vitesse des roues et “vitesse de rotation” la différence de ces vitesses. Lors de l'expérience, nous avons imposé une vitesse de translation constante, ne jouant que sur la vitesse de rotation, notée `vrot`. Les valeurs possibles vont de –10 (complètement sur la gauche) à 10 (complètement sur la droite).
- *Prétraitements sensoriels.* Les valeurs lues sur les proximètres sont résumées en deux variables : `dir`, interprétable comme la direction d'un obstacle, et `prox`, interprétable comme la distance de cet obstacle. Ces variables sont calculées comme suit, v1…v6 désignant les six proximètres avant (de gauche à droite) :

$$\mathtt{dir} = \left| \frac{90\ (\mathtt{v6} - \mathtt{v1})\ +\ 45\ (\mathtt{v5} - \mathtt{v2})\ +\ 5\ (\mathtt{v4} - \mathtt{v3})}{1\ +\ 9\ (\mathtt{v1} + \mathtt{v2} + \mathtt{v3} + \mathtt{v4} + \mathtt{v5} + \mathtt{v6})} \right| \text{ et}$$

$$\mathtt{prox} = \left| \max\ (\mathtt{v1} + \mathtt{v2} + \mathtt{v3} + \mathtt{v4} + \mathtt{v5} + \mathtt{v6})\ /\ 16 \right|.$$

`dir` prend ses valeurs entières entre –10 (obstacle à gauche du robot) et 10 (obstacle à droite), et `prox` va de 0 (obstacle absent) à 15 (obstacle très proche).

La structure de dépendance utilisée est :

$P_C\ (\mathtt{vrot\ dir\ prox}) = P_C\ (\mathtt{dir\ prox})\ P_C\ (\mathtt{vrot} \mid \mathtt{dir\ prox})$,

avec $P_C\ (\mathtt{dir\ prox})$ équiprobable a-priori sur toutes les valeurs possibles.

Une première expérience a été conduite avec une connaissance préalable gaussienne pour caractériser $P_C\ (\mathtt{vrot} \mid \mathtt{dir\ prox})$, de façon similaire au cas de la bassine. Les résultats sont satisfaisants : le robot évite tous les obstacles présents sur son chemin, et souvent même les obstacles mobiles. Il est à noter que le robot est “droitier”, c'est-à-dire qu'arrivant en face d'un obstacle il va de préférence l'éviter par la droite. Cela est dû au fait que les quatre personnes ayant téléguidé le robot lors de la phase d'apprentissage avaient elles-mêmes une nette préférence pour l'évitement par la droite, et cela se retrouve dans les données et donc dans le comportement du robot.

Il existe des situations toutefois où le robot se comporte mal : lorsqu'il rencontre un obstacle assez près et exactement de face, il “s'affole”, hésite sur la direction à prendre et percute l'obstacle. La raison en est que lors de l'apprentissage, dans de telles situations l'opérateur choisissait avec des fréquences comparables d'éviter l'obstacle par la gauche ou par la droite. Les gaussiennes obtenues sont centrées sur zéro avec de grands écarts-types, et le comportement du robot devient quasiment aléatoire dans ce cas.

Nous avons donc recommencé l'expérience avec des distributions non plus de Gauss mais de Laplace. La distribution de Laplace est la distribution de maximum d'entropie correspondant à la connaissance préalable que “l'ensemble des données obtenues proviennent d'une seule et même expérience et peuvent prendre un nombre fini de valeurs” (ce qui est sans doute la connaissance préalable la plus “faible” qui puisse donner lieu à un apprentissage) :

Étant donné une expérience dont le résultat peut prendre K valeurs $a_1 \ldots a_K$, et étant donné N résultats parmi lesquels n_1 ont donné la valeur a_1, n_2 la valeur a_2, …

et n_K la valeur a_K (donc $n_1+\ldots+n_K=N$), alors la probabilité d'obtenir une valeur a_i pour le résultat suivant est donnée par : $p(a_i) = \frac{n_i + 1}{N + K}$ (Jaynes 1994).[8]

Cette modélisation est moins synthétique que la prédédente, puisqu'il faut garder en mémoire l'histogramme complet des résultats obtenus, mais la qualité de l'évitement d'obstacle s'est nettement améliorée[9], surtout dans les situations où un obstacle se trouve exactement en face du robot.

On trouvera décrites dans [Mekhnacha 1995] de nombreuses autres expériences constituant des variations sur ce thème : suivi de contours, orientation vers une source lumineuse en environnement encombré, combinaison de comportements appris séparément, suivi d'objets mobiles, reconnaissance d'objets via un suivi de contour, utilisation d'une fonction d'évaluation pour un apprentissage non supervisé.

---

8 Lorsque N=0 (aucune donnée) on retrouve la distribution uniforme 1/K ; lorsque $N \to \infty$ on retrouve la fréquence expérimentale n_i/N ;et lorsque K=2 on retrouve la célèbre “loi de succession de Laplace”.

9 La qualité d'un évitement d'obstacle est assez subjective. Il est difficile de quantifier les résultats, d'autant plus qu'au delà d'une certaine densité la configuration particulière des obstacles intervient beaucoup (rappelons que la vitesse de translation du robot est constante). Pour des environnements peu encombrés, les obstacles étant blancs et le terrain rectangulaire limité par des murs blancs, le robot heurte très rarement un obstacle ou un mur, et évite encore la plupart des obstacles de couleur bien qu'ils soient détectés moins loin que les obstacles blancs.

CHAPITRE 5

INCRÉMENTALITÉ

L'expérience est une petite lumière
que l'on porte dans le dos et qui
n'éclaire que le chemin parcouru.
Auteur inconnu

La démarche de développement des robots que nous allons décrire et justifier dans ce chapitre est la suivante :

- On programme le robot pour effectuer un certain comportement (typiquement un comportement réflexe ou téléopéré).
- Ce comportement, exécuté dans un environnement particulier, induit certaines dépendances entre variables, qui peuvent alors être apprises. Les structures supportant un tel apprentissage sont appelées “dépendances empiriques probabilistes” (DEP).
- On peut exploiter ces DEP de diverses façons pour faire faire certaines tâches au robot. *Elles peuvent en particulier être utilisées même dans des contextes autres que celui ayant permis l'apprentissage initial.* Les DEP sont en ce sens des “ressources génériques de programmation” pour le concepteur.
- Les DEP une fois apprises peuvent récursivement permettre l'apprentissage de nouvelles DEP.

L'application itérée de cette démarche constitue le “cycle de l'incrémentalité” que nous comptons insérer entre conception et adaptation. L'incrémentalité, dans ce cadre, consiste à développer et exploiter le répertoire de DEP du robot.

Nous définirons dans la section A ce qu'est précisément une DEP. Nous décrirons ensuite dans la section B des expériences robotiques mettant en oeuvre la démarche proposée. Dans la section C nous préciserons quelques caractéristiques de cette démarche.

A. DÉPENDANCES EMPIRIQUES PROBABILISTES (DEP)

Intuitivement, une dépendance empirique est une dépendance apprise expérimentalement entre plusieurs variables du robot. Le terme “dépendance” désigne ici toute relation qui permet, étant donné une contrainte sur une partie des variables, d'avoir une information sur les valeurs des autres variables. Dans la pratique on peut penser à de nombreux types de dépendances : relation fonctionnelle explicite ou implicite, corrélation statistique, détection de coïncidence, périodicité, contraintes semi-algébriques, etc… La liste n'est limitée que par les moyens mathématiques et techniques dont nous disposons.

A.1. Définition

Dans le cadre probabiliste que nous avons choisi, nous définissons plus précisément une *dépendance empirique probabiliste,* abbréviée DEP :

> Soit $\vartheta=\{V_1 \ldots V_n\}$ un ensemble de n variables, avec n>1, et D un ensemble de mesures expérimentales de ces variables prises au cours du temps.
>
> Une DEP est une distribution de probabilité $p_c(V_1 \ldots V_n|D)$ telle que p_c a une expression mathématique paramétrique entièrement identifiée par D, c'est-à-dire qu'une fois D connu, toutes les valeurs numériques de la distribution sont déterminées.

Nous noterons $DEP_c(V_1 \ldots V_n)$ la DEP entre les variables $V_1 \ldots V_n$ définie pour une connaissance préalable C (l'ensemble D étant sous-entendu). S'il n'y a pas d'ambiguïté, nous pourrons nous contenter, selon les cas, d'écrire seulement DEP_c ou $DEP(V_1 \ldots V_n)$.

Le chapitre 4 nous offre déjà plusieurs exemples de DEP : une DEP (`cell angle env`) pour l'expérience de la bassine lumineuse, et une DEP(`vrot dir prox`) pour l'expérience du Khépéra téléguidé. Cependant, tandis qu'au chapitre 4 nous n'avions étudié qu'une seule DEP à la fois, ici nous nous intéresserons plutôt à la construction incrémentale et à l'utilisation combinée de plusieurs DEP.

A.2. Variables primitives

Les variables primitives d'un robot sont les termes de base dans lesquels le robot est capable de caractériser (distinguer) les situations (contextes) dans lesquelles il se trouve — ce que l'on nomme aussi son “état”. Elles constituent donc ce qu'on appelle souvent “l'espace des phases”.

Nous distinguerons, parmi les variables primitives, les “variables sensori-motrices” et les “variables artificielles” :

Variables sensori-motrices

Les variables sensori-motrices sont toutes les variables sensorielles et motrices du robot (capteurs ou moteurs). Dans l'exemple de la bassine lumineuse vu au chapitre précédent, ce sont `angle` (motrice) et `cell` (sensorielle).

Par une analogie commode avec la biologie — mais sans doute abusive —, on peut diviser plus finement les variables sensorielles en trois classes :

- Les *variables extéroceptives* proviennent d'un stimulus extérieur au robot, c'est par exemple le cas de `cell`.
- Les *variables proprioceptives* renseignent sur l'état des actionneurs du robot. Nous en verrons un exemple plus loin.
- Les *variables intéroceptives* renseignent sur l'état interne du robot, par exemple le niveau de charge des batteries. Nous n'en avons pas utilisé.

Notons que les variables sensori-motrices ne sont pas forcément liées à des sources indépendantes. On peut par exemple y trouver à la fois une variable et sa dérivée, ou la fréquence et l'amplitude d'un même signal périodique. D'une source physique telle qu'un un courant électrique, on pourrait ainsi définir comme variables séparées les conversions numériques de l'intensité instantanée, de la tension utile, de la phase, de la température, de la tension dérivée, etc…

Variables artificielles

Le cas de la variable `env` est un peu différent. Sa valeur doit être donnée au clavier par l'opérateur, mais nous ne pouvons pas pour autant la voir comme la variable sensorielle correspondant à l'entrée “clavier”, puisque nous pourrions imaginer de donner d'autres valeurs que `env` par le clavier (par exemple le voltage de la lampe). Nous utilisons `env` dans nos inférences comme si elle provenait véritablement d'une sorte de “capteur de nom d'environnement”. Nous appelons une telle variable une “variable artificielle”.

Notons que dans un modèle explicatif classique, par exemple un modèle polyédrique, les variables considérées sont toutes artificielles : coordonnées des sommets des polyèdres, liens entre sommets qui définissent arêtes, surfaces et volumes, position cartésienne du robot, etc…

Dans le cas particulier d'un robot sans variables artificielles, c'est-à-dire dont toutes les variables primitives sont de nature sensori-motrice, l'espace des phases est appelé “espace sensori-moteur”.

Autres variables (non primitives)

Il existe des variables qui ne sont pas primitives, notamment :

- Les variables qui constituent les termes en lesquels le robot caractérisera les décisions à prendre, par exemple des variables représentant intuitivement des “motivations” ou des “humeurs” (e.g. Maes 1990).
- Les paramètres de modélisation, qui ne servent pas à caractériser un état particulier mais qui quantifient les relations entre variables primitives. Par exemple, les paramètres μ et σ des gaussiennes sont des variables au sens informatique ou mathématique, mais pas des variables primitives.
- Les variables intermédiaires, qui n'interviennent pas dans la description du problème mais servent de résultats intermédiaires permettant de mener les calculs. Par exemple, les indices de tableaux ou les compteurs de boucles utilisés dans les algorithmes.

La notion de variable primitive n'est donc pas liée à une implémentation particulière mais à la façon de formuler un problème, de dire quelles informations nous considérons. Les “variables intermédiaires”, par exemple, sont au contraire liées à l'implémentation et dénotent une façon d'organiser des calculs plutôt que de poser le problème.

B. L'INCRÉMENTALITÉ : ILLUSTRATIONS EXPÉRIMENTALES

Cette section décrit des expériences robotiques, dans le but d'illustrer la démarche incrémentale proposée en tête de chapitre.

Le fil directeur principal de ces expériences est le suivant : *comment une DEP, apprise dans un certain contexte, peut-elle être exploitée dans d'autres contextes ?* Cette question est au coeur de notre approche de l'incrémentalité, et nous y reviendrons longuement par la suite.

La première expérience (§B.1) est un raffinement de la bassine lumineuse décrite au chapitre précédent, la seconde (§B.2) met en oeuvre une caméra face à une mire, et la troisième (§B.3) s'inspire de l'histoire du KitBorg pour exploiter un comportement émergent du robot Khépéra.

Nous reprendrons dans ce chapitre la notation $P_c(\ldots|\ldots)$ du chapitre 4, dans laquelle la présence des données en partie condition est implicite.

B.1. L'expérience de la bassine perturbée

L'environnement de la bassine lumineuse (cf. chapitre 4) est perturbé par des bandes de papier de couleur que nous plaçons sur les bords de la bassine. Une seule bande est utilisée à la fois, mais selon deux caractéristiques : sa position dans la bassine, et sa couleur. La figure 1 montre un exemples de perturbation observée. Le protocole d'exploration est un parcours systématique de toutes les positions dans l'ordre. Les mesures étant longues à faire, les angles sont parcourus non plus tous les 5° mais tous les 9°, ce qui fait 39 valeurs possibles au lieu des 71 considérées au chapitre précédent.

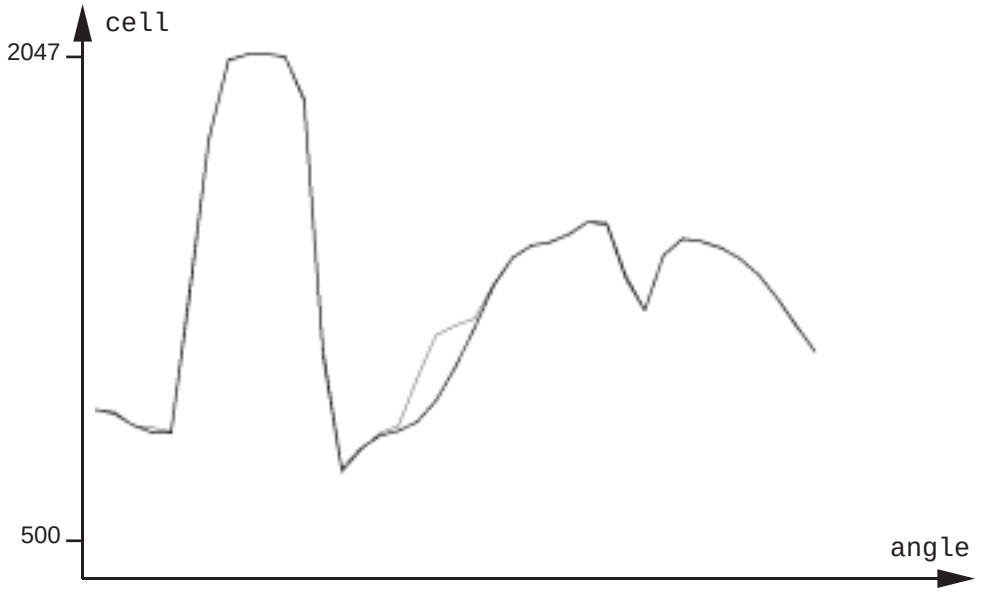


Figure 1. Exemple de perturbation provoquée par une bande de papier (ici rouge) dans la bassine. La courbe en gras représente la moyenne des mesures en l'absence de la bande de papier, et la courbe fine la perturbation provoquée par la bande.

Les bandes de papier sont positionnées à la main autour de marques sur le bord de la bassine. Il y a une assez grande imprécision sur leur centrage autour des

marques, et sur leur inclinaison radiale ou longitudinale (elles glissent facilement sur le fond de la bassine lors du placement). Il y a 7 positions possibles et 3 couleurs.

Nous allons construire incrémentalement deux DEP, qui nous permettront de reconnaître d'abord la position indépendemment de la couleur, puis la couleur elle-même.

Une première DEP (`angle cell`)

La figure 2 montre la DEP (`angle cell`) “de référence” apprise lorsque la bassine n'est pas perturbée. Cette DEP est de même nature que celle décrite au chapitre précédent, à ceci-près que la variable `env` n'y apparaît pas : la modélisation ne prend pas en compte la possibilité de changer d'environnement. Nous appellerons comme d'habitude C la connaissance définissant cette DEP, et `data` les données utilisées.

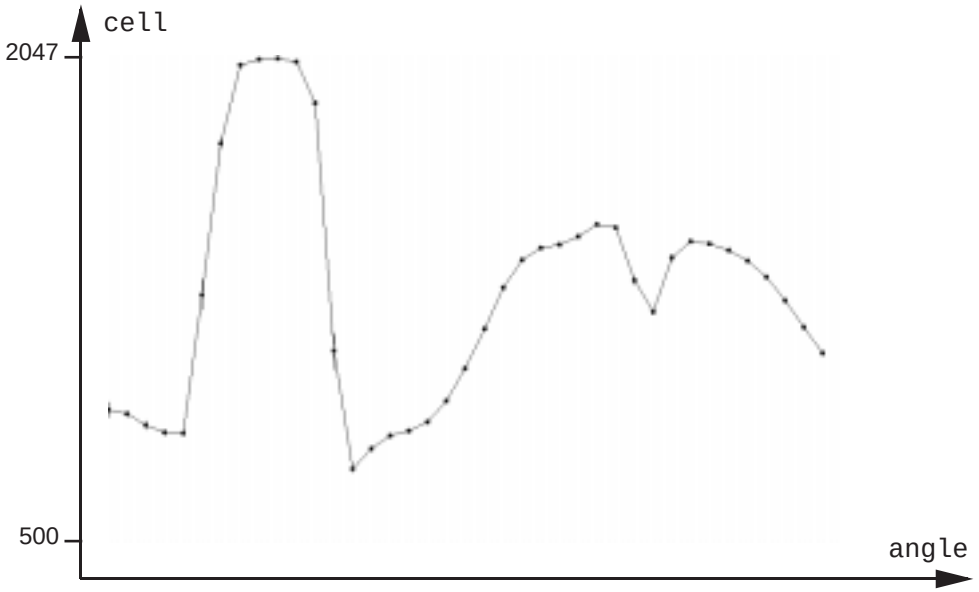


Figure 2. La DEP_C (`angle cell`) de référence, non perturbée. Les barres verticales représentent les écarts-types. Les liaisons entre les points on un rôle purement esthétique, les notions d'interpolation ou de continuité étant absentes du modèle (voir chapitre 4 figure 3).

Variables virtuelles

À présent, plaçons nous dans une situation pouvant être perturbée. Clairement, la DEP ci-dessus n'est plus valable : l'environnement peut changer et dans ce cas les prédictions de la DEP sur la valeur de `cell` ne sont plus vérifiées.

Une façon de gérer ce problème serait par exemple de revoir la modélisation C pour prendre en compte une variable `env` qui distinguerait les environnements, comme nous le faisions au chapitre prédédent.

Toutefois, dans notre optique, la question fondamentale est la suivante :

Peut-on malgré tout utiliser la DEP_C (`angle cell`) *dans ce nouveau contexte ?*

En effet, la valeur P_c (`cell` | `angle`) peut encore être calculée ; certes elle ne permet plus de prédire la valeur lue sur la cellule à une position donnée, mais permet-elle une autre utilisation ?

Le simple fait de poser cette question soulève une ambiguïté de notation. Nous avons deux méthodes pour connaître la valeur de `cell` : soit mesurer le capteur, soit calculer l'inférence P_c (`cell` | `angle`). Dans le contexte exact où cette dernière dépendance avait été apprise les deux méthodes étaient censées donner des valeurs proches. Dans la situation perturbée, elles peuvent se trouver très éloignées, et nous devons leur donner des noms différents. Plutôt qu'une variable sensorielle `cell` et une DDP concernant cette variable, nous allons donc considérer *deux* variables. La notation `cell` reste associée à la variable primitive, la variable inférée quant à elle sera appelée `cell'` (pour rappeler que c'est une grandeur comparable à `cell`). `cell'` est donc la variable qui est inférée via la DEP_c ; elle n'est pas connue de façon déterministe mais par la distribution P_c (`cell` | `angle`). Une telle variable est appelée une *variable virtuelle* :

> Soient des variables $V_1 \ldots V_n$ ayant donné lieu à une DEP $p_c(V_1 \ldots V_n|D)$ dans un certain contexte représenté par l'état de connaissance C et les données expérimentales D. Les variables inférées via cette DEP seront systématiquement distinguée des variables d'origine, le plus souvent en les primant (V'_1) lorsque cela ne soulève pas d'ambiguïté.
>
> Nous appelerons variable virtuelle toute variable ainsi obtenue par inférence via une DEP. Une variable virtuelle n'est toujours connue que par une distribution de probabilités sur ses valeurs possibles.

Si, dans le contexte d'origine ayant permis d'identifier la DEP, V'_1 peut être interprétée comme une prédiction de V_1, dans un autre contexte cette interprétation n'est plus valable. Notons que par contexte nous entendons plus qu'environnement : il s'agit de l'environnement *plus* du comportement observable du robot dans cet environnement. Dans un même environnement, un robot immobile, un robot exécutant un plan, un robot guidé par un réflexe photophile ou un robot téléguidé se sont pas dans le même contexte : ce ne sont pas les mêmes aspects de leur situation qui seront pertinents pour eux.

La figure 3 représente informellement la notion de variable virtuelle.

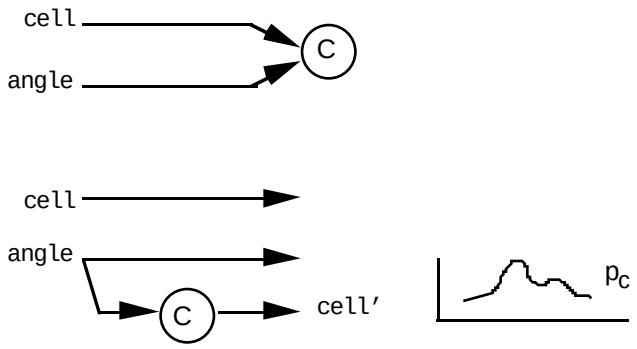


Figure 3. La notion de variable virtuelle. En haut, deux variables `angle` et `cell` forment une DEP_c. En bas, cette DEP_c est utilisée pour inférer `cell` connaissant `angle`. Cette inférence peut donner des valeurs différentes de celles de `cell`, aussi s'agit-il d'une nouvelle variable, notée `cell'` et connue seulement via une DDP.

Une seconde DEP(`place posmax`)

Notre robot dispose donc à présent de ses variables primitives `angle` et `cell`, plus de la variable virtuelle `cell'` (nous n'utiliserons pas ici l'autre variable virtuelle `angle'` que l'on pourrait définir sur la DEP_c). La question est de trouver une utilité à `cell'` malgré le changement de contexte.

Faisons appel à notre imagination… Pour estimer la position de la bande de papier perturbatrice, comparons les valeurs de `cell` lues sur le capteur et les valeurs de `cell'` déterminées par la DEP_c. Il y a fort à parier que la position (`angle`) où les deux valeurs diffèrent le plus correspond assez bien avec la position de la bande ; voir par exemple la figure 4, qui superpose les courbes des moyennes obtenues sans perturbation ou avec chacune des trois couleurs possibles (noir, bleu, rose).

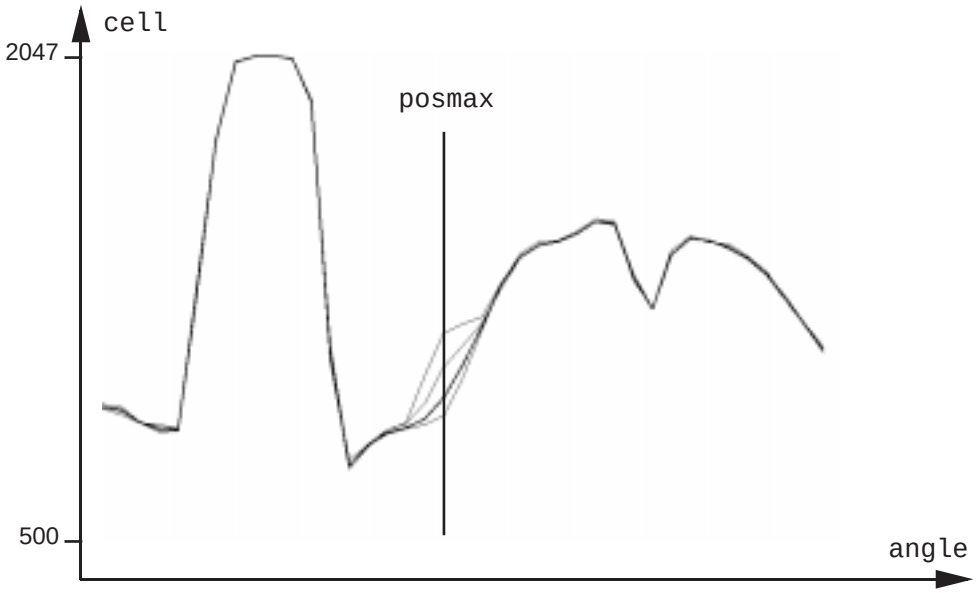


Figure 4. La position de la bande de papier est bien caractérisée par la position `posmax` où on observe un écart maximal entre la courbe non perturbée (en gras) et la courbe perturbée (ici trois courbes, une pour chaque couleur possible).

La position où les deux valeurs diffèrent le plus est calculable via un traitement effectué à partir des variables primitives et virtuelles. Nous pouvons par commodité noter le résultat comme une variable intermédiaire `posmax` (position du maximum), égale à la valeur de `angle` maximisant celle de |`cell`–`cell'`| (qui dépend de `angle`). Ce type d'opération sera génériquement appelé *prétraitement,* de même que la variable intermédiaire désignant la valeur obtenue, ici `posmax`. Notons que ce prétraitement requiert une action et possède un aspect temporel : il faut balayer toutes les valeurs de `angle` avant de déterminer `posmax`. Un prétraitement n'est donc pas forcément, à notre sens, une opération seulement passive (comme prendre le logarithme ou la dérivée d'une variable).

Procédons à présent de la façon suivante : autorisons-nous à fournir au robot via le clavier une variable artificielle `place` désignant la position de l'objet — ceci en vue de tester expérimentalement notre hypothèse selon laquelle `posmax` devrait lui correspondre. Nous allons alors apprendre une nouvelle DEP sur des données $\texttt{data}_2$, adoptant une modélisation S disant que `posmax` dépend de `place` par une gaussienne et que l'a-priori sur `place` est l'équiprobabilité. Cette DEP sera notée DEP_S (`place posmax`), et est schématisée figure 5. La notation allégée $P_S(\ldots|\ldots)$ sera utilisée pour dénoter $p_S(\ldots|\ldots\ \texttt{data}_2)$.

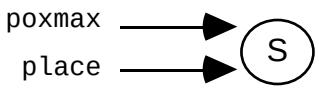


Figure 5. La DEP_S entre `posmax` (prétraitement impliquant une série de mesures) et `place` (variable artificielle).

Il apparaît une difficulté dans cette utilisation du prétraitement `posmax` : `cell'` n'étant connu que via une DDP, en toute rigueur |`cell`–`cell'`| l'est aussi, et donc la position du maximum également. Ceci introduit des problèmes de calcul très importants :

- il faut expliciter la DDP sur `posmax` en fonction des variables primitives,
- il faut faire intervenir cette DDP dans le calcul effectif de `place'`,
- la méthode d'apprentissage de la DEP_S devient très problématique.

Pour ces raisons, nous somme obligés d'éliminer à chaque étape le caractère probabiliste des variables virtuelles utilisées, par un processus de décision. Dans notre cas, avant d'utiliser `cell'` pour le calcul de `posmax`, nous la réduirons à sa valeur la plus probable, ici la valeur du paramètre μ(`angle`, `data`) qui identifiait les dépendances gaussiennes de `cell` dans la DEP_C. Nous sommes alors ramenés à des manipulations simples, le prétraitement `posmax` devenant déterminé et non plus probabiliste. Mais ce faisant nous abandonnons provisoirement l'idée de formaliser la manipulation *globale* du système de DEP.[1] Cette opération est schématisée figure 6.

---

[1] Le maintien d'une exactitude globale des calculs, et donc de la consistance formelle du système d'inférence, ne semple pas envisageable dans le cas général. Une attaque plus “propre” de ce problème consisterait à inventer un système d'inférence de portée restreinte (et donc incomplet) mais dans lequel l'ajout d'une DEP pourrait néanmoins se faire de façon consistante. Il semble que certains travaux sur les réseaux bayésiens

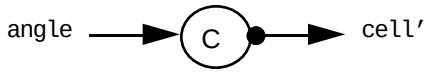


Figure 6. Une variable virtuelle n'est connue que via une DDP, mais afin de manipuler des variables toutes déterminées nous avons choisi de prendre systématiquement la valeur la plus probable. Cette opération est schématisée par le petit cercle noir. Des solutions alternatives seraient de prendre l'espérance de la DDP, ou encore de tirer la valeur selon la loi de cette DDP (traiter une variable virtuelle comme une “variable aléatoire” au sens classique).

Ayant donc fait ce choix de prendre systématiquement la valeur la plus probable de `cell'` pour calculer une valeur déterministe de `posmax`, nous avons mené l'expérience et obtenu la DEP_s (`place posmax`) montrée figure 7.

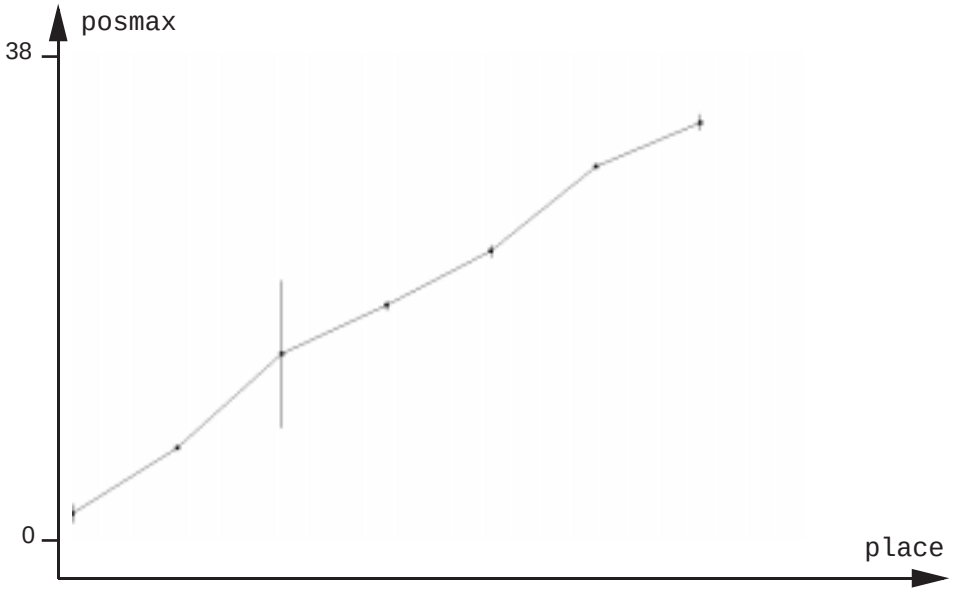


Figure 7. La DEP_s(`place posmax`). Les liaisons entre les points on un rôle purement esthétique, le modèle ne faisant aucune sorte d'interpolation. La seconde position est difficile à déterminer (grand écart-type), car elle se situe dans une zone d'ombre faisant suite à une zone très éclairée (voir figure 8c un peu plus loin). L'influence des bandes de papier est faible dans la zone sombre, mais une petite imprécision dans le placement de la bande peut affecter la zone claire et provoquer des perturbations plus grandes ailleurs.

Cette seconde DEP permet de se passer par la suite de la variable artificielle `place` et de l'inférer plutôt par `place'`, via P_s (`place | posmax`).

Pourquoi une variable virtuelle et non une “estimation” ?

Une fois arrivés là, nous pouvons alors nous demander : était-ce bien nécessaire ? En effet, pourquoi ne pas directement utiliser `posmax` comme estimation de la position de la bande de papier, plutôt que `place'` ?

---

puissent être vus comme l'embryon d'une telle démarche, mais ils reste encore beaucoup à accomplir (Pearl 1991, ch. 8, “learning structure from data”, où on décrit la formalisation d'une forme d'incrémentalité limitée).

La distinction est essentielle. En passant par la DEP, l'incertitude déterminée expérimentalement est prise en compte. Par exemple, lorsque `place'` indique la position numéro 2, elle indique également qu'il ne faut pas accorder une importance capitale à ce résultat car il est très incertain. Par contre, utiliser `posmax` directement comme représentant la position reviendrait à faire l'interprétation a-priori "`posmax` correspond à la position de la bande de papier", et donc à considérer `posmax` comme un terme prédéfini.

L'expérimentation n'est donc pas à voir comme une "validation" de la méthode d'estimation de la variable artificielle, ce qui conduirait à utiliser `posmax` *à la place* de `place`, et donc à ne pas tenir compte des incertitudes identifiées. De plus on ignorerait ainsi la relation à une situation concrète connue (l'expérience effective ci-dessus), qui seule permet de guider l'interprétation de `place'` et peut l'amener à se distinguer de celle (prédéfinie) de `place`. *C'est là le rôle principal de l'expérimentation : fournir un contexte concret permettant de guider l'interprétation du concepteur, et sortir ainsi de l'ornière des représentations purement abstraites.*

Nous reviendrons au §C.4 sur ce rapport entre les notions d'incertitude et d'interprétation.

Une troisième DEP(`couleur posmax valmax`)

Nous n'allons pas arrêter là notre démarche incrémentale. Il reste des aspects de la situation que nous n'avons jusque là pas pris en compte : par exemple, les perturbations provoquées pour obtenir les données expérimentales $\texttt{data}_2$ n'étaient pas seulement localisées, mais colorées. Y'aurait-il moyen d'exploiter à présent les deux DEP présentes pour déterminer la couleur ?

Là encore nous devons utiliser un minimum d'imagination, et penser que si la position est bien représentée par la commande ayant permis de maximiser l'écart entre `cell` et `cell'` (prétraitement `posmax`), cet écart lui-même peut être (pour une position donnée) un bon indicateur de la couleur ayant été utilisée. Les figures 8a, 8b et 8c étayent cette intuition (voir également la figure 4).

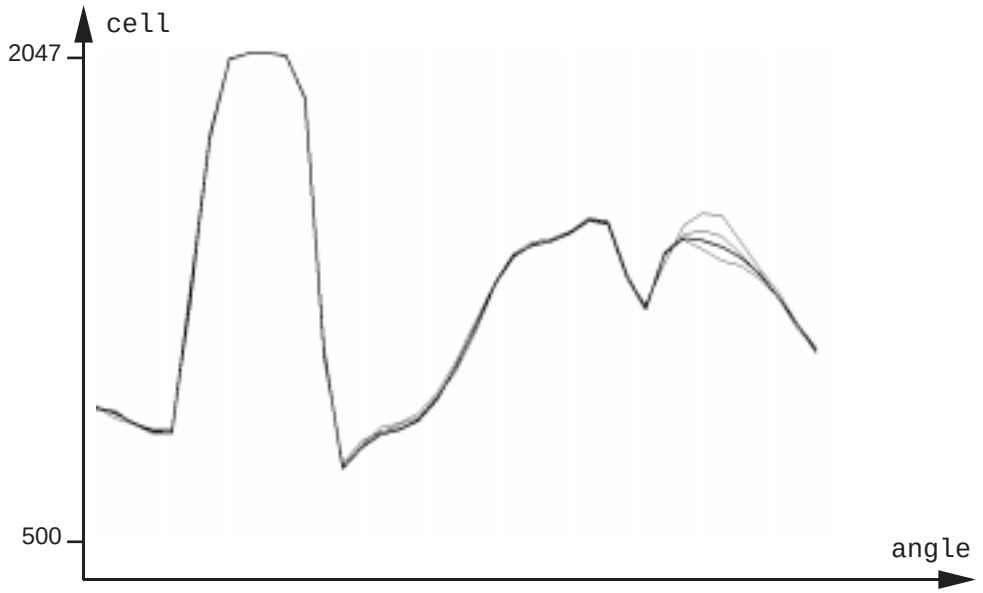


Figure 8a. Perturbations provoquées par des bandes de couleurs différentes placées à la même position. La courbe de référence est en gras. Les courbes fines correspondent de haut en bas aux bandes rose, bleue et noire. La différence avec la courbe de référence semble un bon indicateur de la couleur.

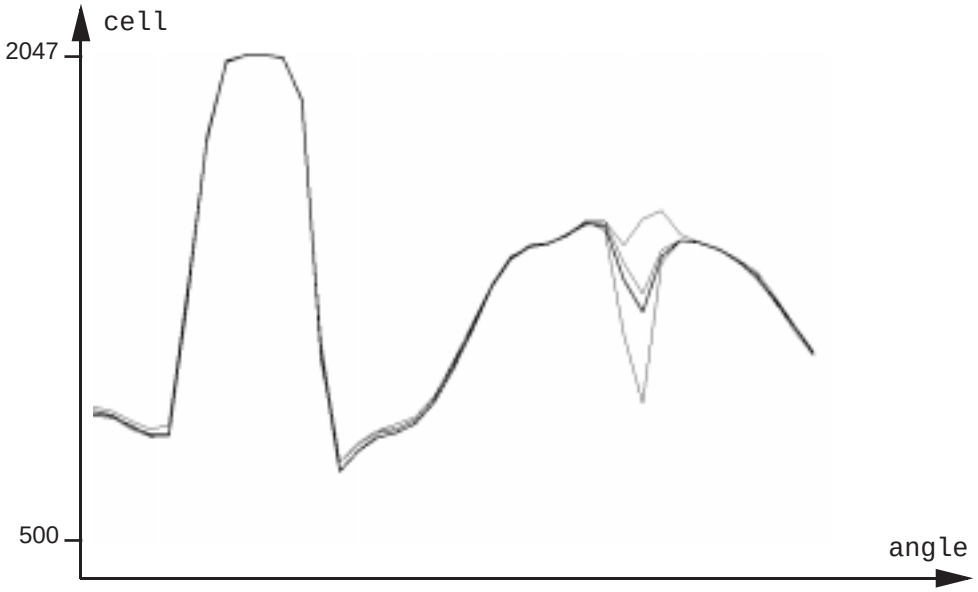


Figure 8b. Un second exemple, à une position différente.

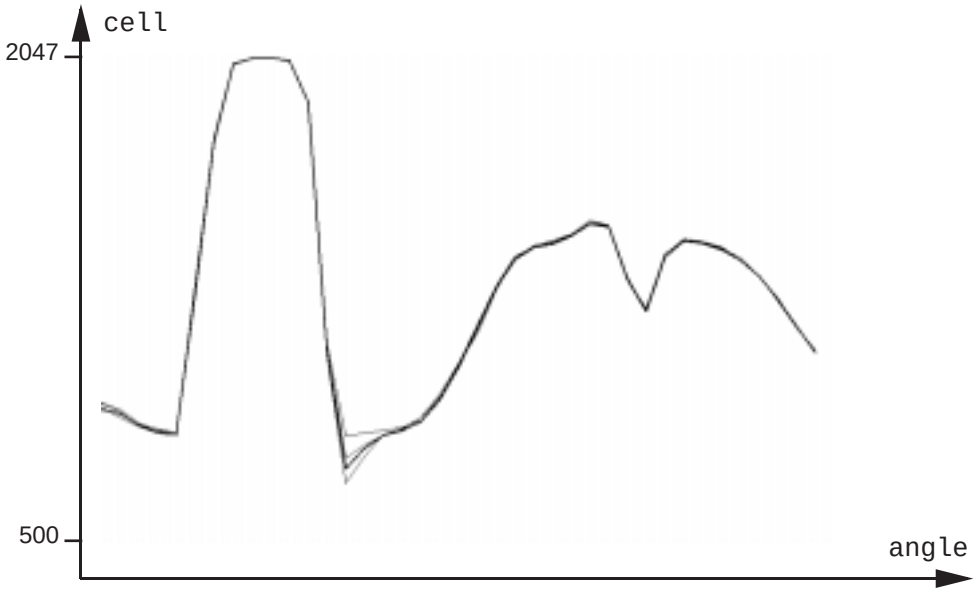


Figure 8c. Un troisième exemple, à une position différente (il s'agit de la position numéro 2, dont la détermination est difficile via la DEP_S, voir figure 7).

Nous pouvons alors sortir l'écart maximum comme un prétraitement, obtenant une variable intermédiaire `valmax` (valeur du maximum) égale à la différence `cell'–cell` déterminée à la position `posmax`. Nous déterminons encore `cell'`, variable virtuelle normalement probabiliste, par sa valeur la plus probable. Les prétraitements `valmax` et `posmax` sont alors résumés sur la figure 9.

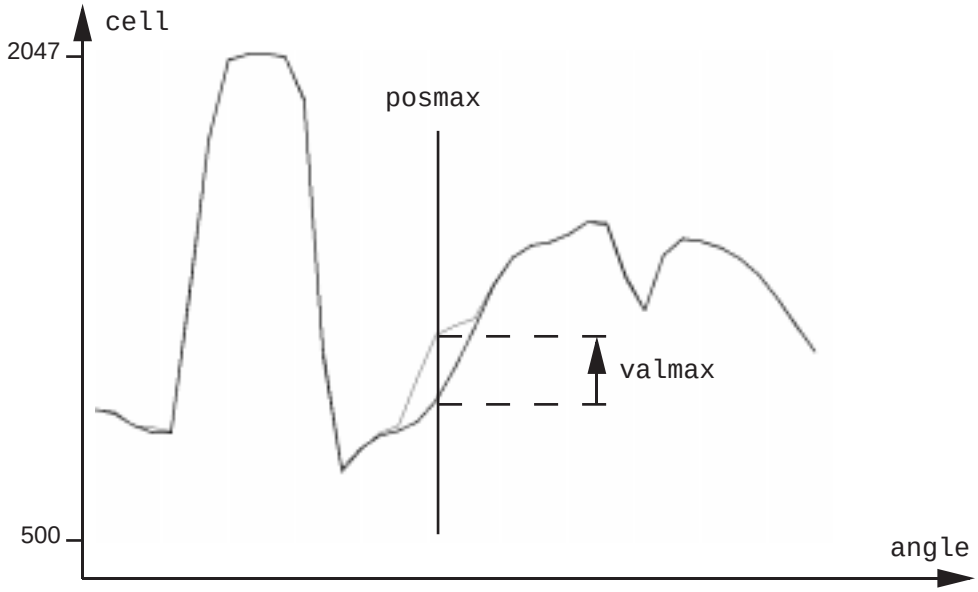


Figure 9. Les prétraitements posmax et valmax.

De la même façon que nous avions introduit une variable place pour superviser l'apprentissage de la position, nous utilisons de même une variable artificielle couleur pour désigner la couleur. Nous optons encore pour un modèle du même type que les précédents : valmax dépend de couleur par une gaussienne, et couleur est a-priori équiprobable, connaissance que nous appelons U. L'expérience nous a fourni les données $data_3$, et nous avons identifié une DEP_u (couleur valmax posmax) montrée sur la figure 10. On peut alors ne plus considérer couleur et retrouver la couleur par inférence via cette troisième DEP.

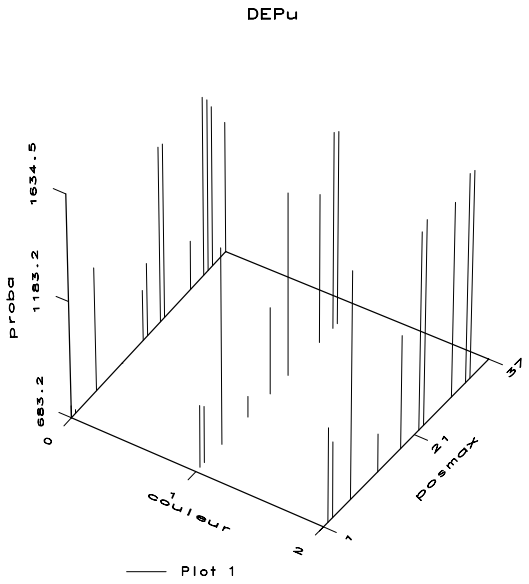


Figure 10. La DEP_u(couleur valmax posmax). Sur cette figure, valmax n'est pas calculée comme sur la figure 9, mais est égale à la valeur de cell à la position posmax. Seules les espérances de valmax sont montrées, en fonction des deux autres variables. À certains endroits ces espérances ne sont pas définies, aucune donnée expérimentale n'ayant été observée avec ces configurations.

Une autre possibilité est de considérer non pas `posmax` mais la valeur la plus probable de `place'` inférée via la DEP_S, c'est-à-dire la position où se situe probablement la bande (c'est une solution plus économique : il y a 7 valeurs possibles pour `place'` au lieu de 39 pour `posmax`). On obtient alors la DEP_V(`couleur valmax place'`) de la figure 11.

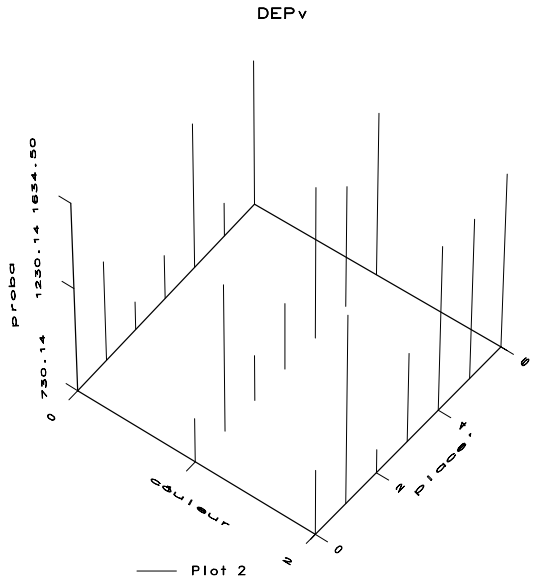


Figure 11. La DEP_V(`couleur valmax place'`). Sur cette figure, `valmax` n'est pas calculée comme sur la figure 9, mais est égale à la valeur de `cell` à la position `posmax`. Seules les espérances de `couleur` sont montrées, en fonction des deux autres variables.

Utilisation des trois DEP pour un “bricolage” instructif

Peut-on, avec les DEP à présent identifiées, fournir au système la position de la bande de papier et obtenir qu'il détermine expérimentalement la couleur ? L'enjeu de cette question est le suivant. Jusqu'ici, nous utilisions la même action (balayage) pour déterminer les deux prétraitements et donc à la fois position et couleur ; or la question posée à présent demande justement ne pas tout balayer mais de profiter d'une information qui évite ce travail. Rigoureusement, la modélisation utilisée (C,S,U,V) ne prend pas en compte cette possibilité. Or, nous pouvons malgré tout, en “bricolant”, répondre à la question sans créer de nouvelles DEP. Voici le “bricolage” en question, schématisé aussi sur la figure 12 :

- La valeur *`place`* de `place` étant fournie, nous utilisons la DEPs pour obtenir une valeur *`posmax'`* de `posmax`'. Intuitivement, *`posmax'`* prédit la position autour de laquelle l'écart maximum entre `cell` et `cell`' se situe.
- Commandons alors au robot d'aller mesurer `cell` à cette position *`posmax'`*. Cela nous donne une mesure *`cell`.*

- Dans une bassine non perturbée, nous nous serions attendus à trouver non pas *cell* mais la valeur de cell' inférée via P_c (cell | angle=*posmax'*). Prenons-en alors la valeur la plus probable *cell'*.
- Entre *cell'* et *cell* la différence est δ=*cell'*–*cell*. Au vu de cette différence, nous pouvons calculer P_u(couleur | posmax=*posmax'*, valmax=δ). Cette DDP, que nous pouvons appeler couleur', est considérée comme la prédiction désirée quant à la couleur de la bande de papier.

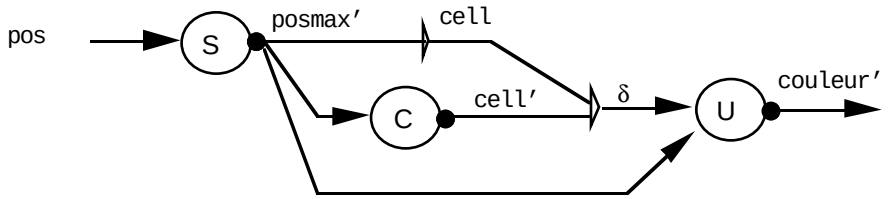


Figure 12. Comment combiner les inférences pour déterminer la couleur sachant la position place. Les ronds noirs indiquent que l'on prend la valeur la plus probable d'une DDP, et les triangles indiquent un prétraitement des valeurs obtenues.

L'aspect fortement “bricolé” de cette démarche, que nous n'avons pas expérimentée, se manifeste clairement dans la question du “sens” que l'on peut attribuer aux incertitudes obtenues sur couleur' : la notion d'incertitude utilisée jusque là dans nos exemple était bien propre, bien assise théoriquement sur le chapitre 3, mais le manque de cohérence formelle dans la façon dont nous avons utilisé les trois DEP pour obtenir couleur' est bien plus douteuse. La réponse à cette question est repoussée au §C.4.

Malgré ses difficultés, ce genre de bricolage illustre bien l'idée centrale que les DEP puissent être “réinterprétées” de multiples façons dans différents contexte. Il n'existe pas une façon unique d'exploiter une DEP pour programmer un robot.

Résultats : reconnaissances de positions et de couleurs

Voici les résultats quantitatifs obtenus pour les manipulations précédentes.

L'ensemble d'apprentissage complet pour DEP_s, DEP_u et DEP_v ne comporte que 134 exemples : 6 séries de mesures pour chaque couple (*place, couleur*), plus une série partielle interrompue. C'est peu, mais l'expérimentation est longue est dangereuse : pour chaque mesure nous enlevons et remplaçons la bande de papier à la main, afin de tenir compte des incertitudes de ce placement, et pour cela nous devons nous aventurer à portée du bras de robot, qui peut “partir” brutalement pour des raisons mal connues.

Sur l'ensemble d'apprentissage pris comme ensemble de test, place' manque deux fois de trouver la bonne position, et il s'agit toujours de la position numéro 2, dont l'incertitude avait été reconnue comme grande (figure 7). couleur' inférée via DEP_u se trompe quant à elle une seule fois, et 8 fois en utilisant DEP_v.

En réservant entre 35 et 40 exemples (25 à 30%) comme ensemble de test et en gardant les autres pour l'apprentissage, nous avons toujours obtenu 0 ou 1 erreur sur la position (dont une fois sur la position 3 et non 2), 5 ou 6 sur la couleur via DEP_u, et 4 ou 5 sur la couleur via DEP_v. Environ la moitié des erreurs sont

communes à DEP_u et DEP_v, et autrement les deux se trompent aussi souvent l'une que l'autre. Cela fait très grossièrement autour de 15% d'erreur sur la couleur (des chiffres précis ne voudraient pas dire grand chose avec aussi peu d'exemples). Notons que lorsque la position elle-même est bonne, il n'y a jamais d'erreur pour déterminer la couleur rose, qui se distingue très bien des noire et bleue.

DEP_u répond parfois (0 à 2 fois) “indéfini” pour `couleur'` : c'est lorsque la valeur mesurée de `posmax` n'a jamais été rencontrée lors de l'apprentissage. En effet, notre modélisation ne fait aucune sorte d'interpolation entre valeurs angulaires, et à une position jamais rencontrée les paramètres des gaussiennes sont donc indéterminés (et il n'y a pas assez d'exemples pour assurer que toutes les positions auront été rencontrées). Une réponse “indéfini” est comptée comme une erreur.

Même si les performances sont comparables en termes d'erreurs, la DEP_v est plus économique que la DEP_u et fournit toujours une réponse, et semble donc plus intéressante.

B.2. L'expérience de la mire

L'expérience de la mire est dans son principe similaire à la précédente, mais se situe dans un contexte sensori-moteur totalement différent. Le dispositif est montré figure 13. Une caméra est commandée en translation, approximativement le long de l'axe optique (positionnement à la main sans calibration). La variable de commande est notée `trans`. En face de la caméra se trouve une mire fixe. Lorsque le robot (la caméra) bouge, les modifications de l'image sont entièrement dûes à ce mouvement. Le premier stade de l'expérience est d'apprendre empiriquement la dépendance entre le mouvement et les modifications de l'image.

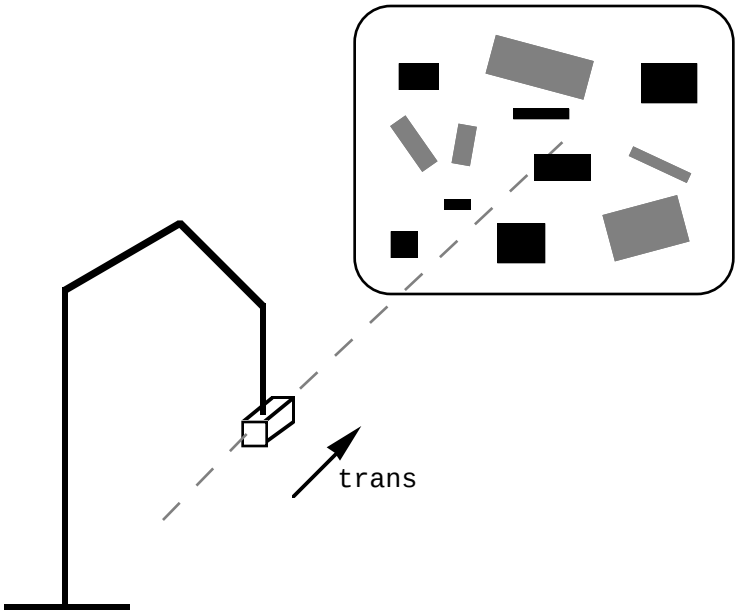


Figure 13. Le dispositif pour l'expérience de la mire. Une caméra, montée sur un bras de robot, se déplace face à une mire (un degré de liberté).

L'outil de base pour caractériser ces dernières est un programme de mise en correspondance d'indices visuels (Boufama 1994) effectuant un traitement entre

deux images d'une même scène prises de deux positions différentes. Le résultat obtenu est un ensemble A de couples ((x,y), (x',y')) de points en correspondance entre les deux images, comme décrit figure 14. La variable sensorielle primitive du robot, qui est l'image (tableau de pixels), est toujours prise en compte via ce traitement préalable, et nous considèrerons pour simplifier que A (les correspondances) est elle-même une variable primitive.

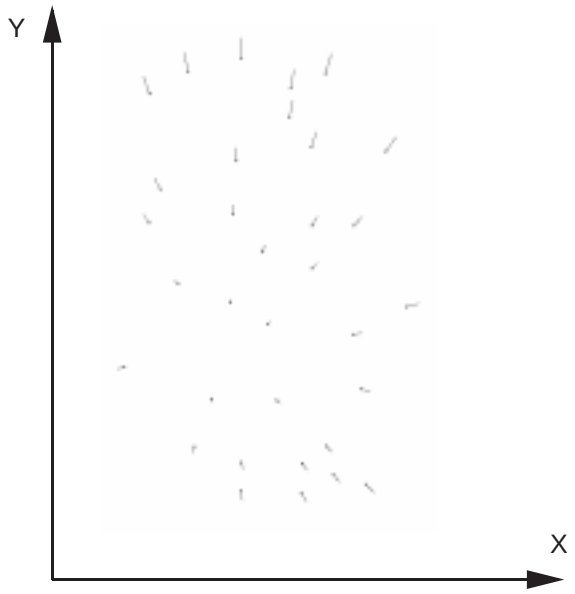


Figure 14. Un exemple de correspondance. Les gros points sont les positions dans l'image avant le mouvement, et les extrémités des segments partant de ces points sont les positions correspondantes après le mouvement. L'exemple ici provient d'une translation vers l'avant.

DEP entre le mouvement et la correspondance visuelle

Comme nous l'avons dit, le premier problème est d'apprendre empiriquement une DEP entre le mouvement `trans` et les modifications de l'image. Pour cela nous allons essayer deux méthodes :

- une méthode intuitive, qui consiste à se dire que la translation de la caméra pourrait être corrélée avec l'écart-type moyen des variations des points de l'image le long de l'axe X (axe horizontal sur l'image).[2] Notons alors la variable (prétraitement) `écart` = $n^{-1} \sum_i (x'-x)^2$, i=1…n, n étant le nombre de correspondances dans A.
- une méthode analytique, qui consiste à modéliser la situation dans les termes de la géométrie projective pour obtenir la grandeur caractérisant ce mouvement. Ceci est effectué en annexe 2, certaines hypothèses simplificatrices étant nécessaires pour résoudre les calculs. Le résultat est la variable (prétraitement) `rapport` = $n^{-1} \prod_i (y'-y°)/(y-y°)$, i=1…n, n étant le nombre de correspondances dans A, et y° l'ordonnée du centre de l'axe optique sur l'image (point invariant en translation). Cette formule combine donc des rapports d'homothétie. Le point y° est déterminé

[2] Nous aurions pu utiliser également l'axe Y, ou les deux axes. Mais certaines expériences comportaient des mouvements latéraux influant fortement sur les correspondances en Y (la caméra étant montée à 90° sur le robot), et c'est pour cela que c'est l'axe X qui a été réservé pour caractériser les mouvements longitudinaux).

expérimentalement comme décrit en annexe 2 ; l'imprécision est importante mais son influence est négligeable dans la pratique : nous avons essayé toutes les valeurs de y° dans l'intervalle d'incertitudes, sans conséquences visibles.

Nous allons décrire toute l'expérience avec la variable de prétraitement `écart`, qui a en fait toujours donné de meilleurs résultats que `rapport`.

La première étape est de tester notre hypothèse : `écart` permet-elle une bonne caractérisation de `trans` ? Pour cela nous allons encore faire une hypothèse gaussienne C telle que P_C (`écart` | `trans`) = G(`écart` | μ(`trans`, `data`), σ(`trans`, `data`)), `trans` suivant une loi équiprobable. Nous identifions alors μ et σ expérimentalement par une série de mouvements partant tous de la même distance initiale à la mire, distance non prise en compte dans la modélisation C mais que nous noterons δ°. Nous obtenons ainsi une DEP_C (`écart trans`) montrée figure 15.

Note importante: les résultats présentés dans les figures 16 à 20 incluent des mouvements latéraux de la caméra. Pour des raisons techniques (vibrations importantes du robot) nous n'avons pas pu reprendre effectivement l'expérience plus simple décrite ici, et les seuls enregistrements antérieurs disponibles comportaient ces mouvements latéraux. L'idée à l'origine était de les faire également inférer par une DEP supplémentaire, ce qui marche bien mais ajoute un bruit important aux autres résultats de l'expérience.

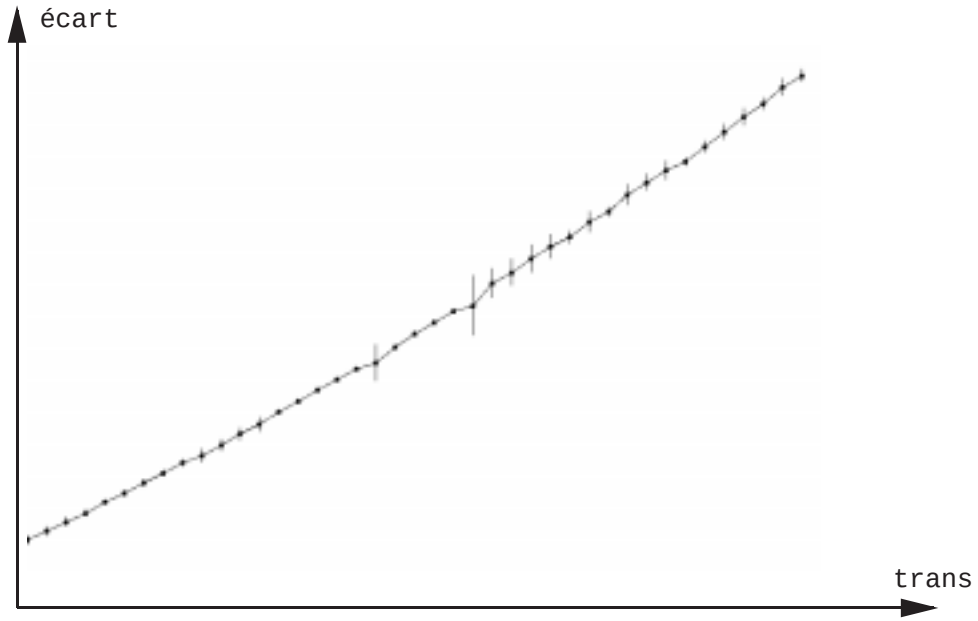


Figure 15. La DEP_C (`écart trans`). Les barres verticales représentent les écarts-types (voir chapitre 4 figure 3). Les mouvements `trans` effectués vont de -20 à 20 centimètres, avec un pas de 1 cm. Ces mesures incluent des déplacements latéraux parasites, de -5 à 5 centimètres.

Cette DEP permet entre autres d'inférer `trans` à partir de `écart`, et donc de réaliser l'expérience test suivante. La mire étant placée initialement à une distance δ°, l'opérateur l'amène à la position δ°+x, x entre -20 et 20 centimètres. En utilisant la DEP_C, le robot a alors “l'impression visuelle” d'avoir bougé de -x, et donc nous

pouvons considérer –`trans'` comme une estimation de x (notons une fois encore la différence fondamentale entre `trans`, qui dans ce test est nulle, et `trans'` inférée via la DEP_c). Exploitons alors cette idée pour développer le système un peu plus.

Seconde étape : prise en compte de la distance

Ayant résolu le problème initial de trouver une DEP entre mouvement et correspondance visuelle, nous trouvons gênant que cette DEP ait été développée pour la distance spécifique $\delta°$. L'idée est alors, plutôt que de recommencer la modélisation en tenant compte du facteur distance, de chercher à exploiter notre première DEP pour aider à gérer le cas des autres distances.

Que se passe-t-il si nous utilisons cette DEP_c pour inférer `trans` à partir de `écart` dans une situation où la distance de la mire n'est pas $\delta°$ mais δ ? Si $\delta>\delta°$, les variations sur l'image seront plus faibles que si la distance avait été $\delta°$, et la variable inférée `trans'` proposera donc une sous-évaluation de `trans` effectivement commandée. Si $\delta<\delta°$ c'est l'inverse, `trans'` sur-évalue `trans`.

L'idée est qu'un critère de comparaison entre ces deux valeurs, un prétraitement `illusion`, pourrait caractériser la distance. Nous adoptons alors une variable artificielle `distance` pour tester cela. Nous prendrons `illusion = trans'/trans`, intuitivement plus pertinent que `trans'–trans` (et qui dans le cas de l'utilisation de la variable `rapport` peut être demontré analytiquement comme le prétraitement idoine).

Nous cherchons alors à apprendre une DEP_s (`distance illusion`), toujours de nature gaussienne (S modélise `illusion` comme dépendant de `distance` de façon gaussienne, `distance` étant par ailleur équiprobable). L'ensemble des DEP est schématisé figure 16.

Les résultats obtenus ici pour S sont mauvais, à cause des déplacements latéraux parasites dans notre ensemble de données.

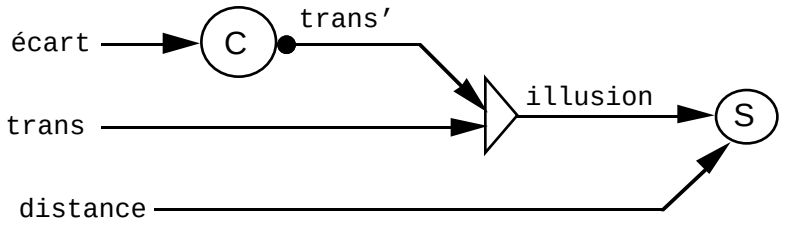


Figure 16. L'ensemble des DEP et des prétraitements pour l'expérience de la mire.

Exploitation de DEP_s

Nous avons trouvé deux applications à la DEP_s :

- Détermination de la distance d'une mire : la mire étant placée (à la main) en face de la caméra, le robot tente par des mouvements de déterminer sa distance.
- Suivi de mire : le robot essaye de maintenir la caméra à une distance constante de la mire. Lorsque l'opérateur translate la mire, les modifications de l'image permettent au robot d'estimer ce mouvement, et de le compenser par son propre mouvement.

Résultats des expériences

Les expériences n'ayant pu être refaites pour chiffrer les résultats de façon systématique (cf note plus haut), nous ne pouvons citer que les souvenirs qui nous restent des résultats empiriques que nous avions obtenus à l'époque.

La première application (détermination de la distance d'une mire) avait été programmée ainsi : nous choisissions une certaine valeur x de déplacement, par exemple 15 cm. La caméra avançait alors de 15 cm puis reculait d'autant (ce qui la qui ramenait donc à la position initiale). Les estimations pour chacun de ces mouvements étaient justes à ± 4 centimètres, pour estimer des distances variant tous les 2 cm entre -20 à 20 centimètres de la distance de référence $\delta°$. Les valeurs pour x étaient prises entre 12 et 18 cm, car les valeurs plus petites n'étaient pas assez significative pour donner de bonnes estimations, et les valeurs extrèmes provoquaient souvent des aberrations (dûes à des mises en correspondances difficiles entre images trop différentes).

Pour combiner un ensemble d'estimations indépendantes effectuées pour une même distance, la multiplication des DDP s'est révélée une mauvaise politique. Les distributions gaussiennes présentent un inconvénient pratique majeur lorsqu'il s'agit de les multiplier : les probabilités extrèmement faibles des valeurs loin de la moyenne (typiquement 10^{-50} à 10^{-250}!) provoquent des instabilités numériques qui faussent très vite les résultats. Il suffit alors d'un résultat aberrant (i.e. peu probable) pour faire disparaître par multiplication un “pic de probabilité” existant autour de la valeur correcte de la distance. Le mieux, dans notre cas, était de rassembler les prédictions faites sur une seule mesure, et de choisir la plus fréquente comme estimation finale.

Une amélioration de la procédure (que nous n'avons pas pu tester) pourrait être de déterminer automatiquement le mouvement le plus informatif dans l'ignorance de la distance à estimer. Ce mouvement une fois effectué donnerait une première estimation de la distance, et on pourrait grâce à cela déterminer le second mouvement à essayer pour confirmer cette première estimation.

La seconde application proposée, qui était de faire suivre automatiquement au robot les mouvements de la mire, avait donné des résultats satisfaisants, avec de temps en temps quelques mouvements aberrants. Là aussi, on pourrait imaginer d'exploiter le mouvement effectué (pour suivre la mire) pour estimer l'erreur commise et améliorer le suivi par un second mouvement correctif.

Un épisode marquant de l'expérience de la mire

L'expérience de la mire a révélé un défaut majeur dans une fonctionnalité de base du bras de robot utilisé (un capteur proprioceptif). D'autres applications plus classiques avaient utilisé aveuglément cette fonctionnalité sans même se douter de l'existence d'un problème. Nous voyons là une illustration flagrante des difficultés que provoque l'adoption de représentations prédéfinies en robotique.

Le bras de robot supportant la caméra offre la possibilité de lire passivement les positions articulaires *effectivement* atteintes à un moment donné, par une variable proprioceptive notée `proprio`. Les variables motrice `trans` et proprioceptive `proprio` peuvent prendre des valeurs différentes dans le cas d'une commande spécifiant des

positions angulaires inatteignables (butées mécaniques), ou bien lors de petits mouvements de l'ordre du millimètre (où des frottements et des phénomènes d'inertie rendent le bras peu précis), ou encore tant que l'exécution de la commande n'est pas terminée.

Notre première idée avait été d'utiliser, pour créer nos DEP, non `trans` mais `proprio`. Nous avons cherché à construire une DEP (`proprio écart`), et avons relevé comme données expérimentales D l'ensemble de couples (*`proprio,écart`*) montré sur la figure 17.

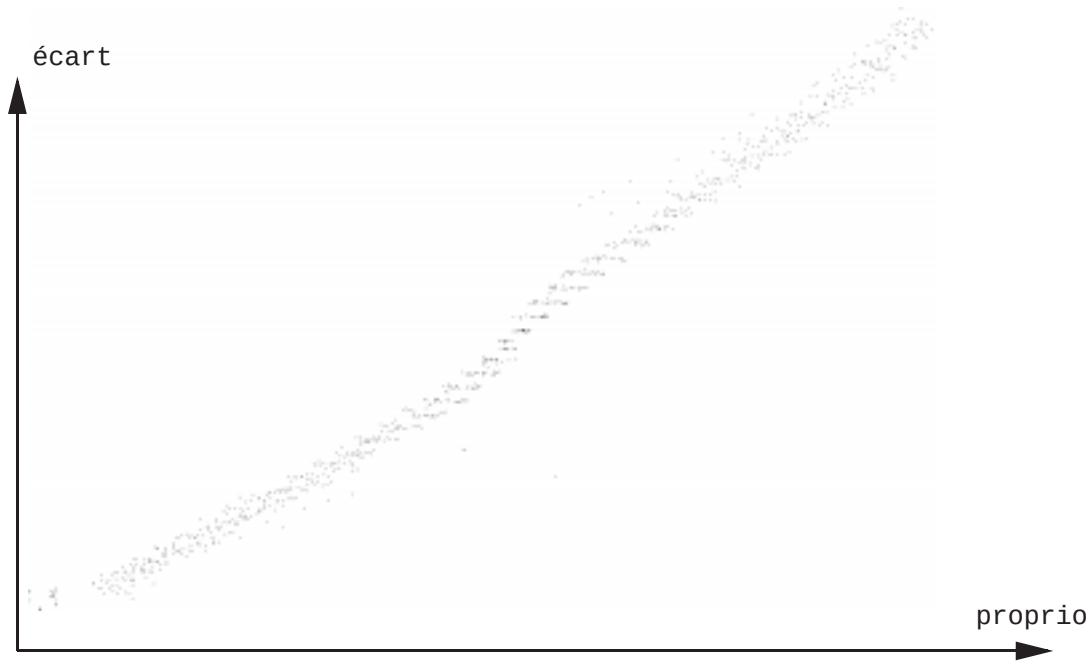


Figure 17. Les mesures de points (*`proprio, écart`*). Le prétraitement visuel sépare assez bien les mesures du mouvement, ce qui est très étonnant (voir texte). Ces mesures incluent des mouvements latéraux faibles (entre -5 et 5 cm), qui contribuent à l'incertitude sur le traitement visuel.

Tout se passe dans la figure 17 comme si les mouvements `proprio` étaient dispersés autour des prétraitements visuels `écart`, ce qui nous a profondément surpris, car nous nous attendions plutôt à ce que ce soit `proprio` qui sépare les mesures de `écart`.

Nous avons alors utilisé `trans` à la place de `proprio`, et obtenu la figure 18, qui correspond plus à ce que nous escomptions.

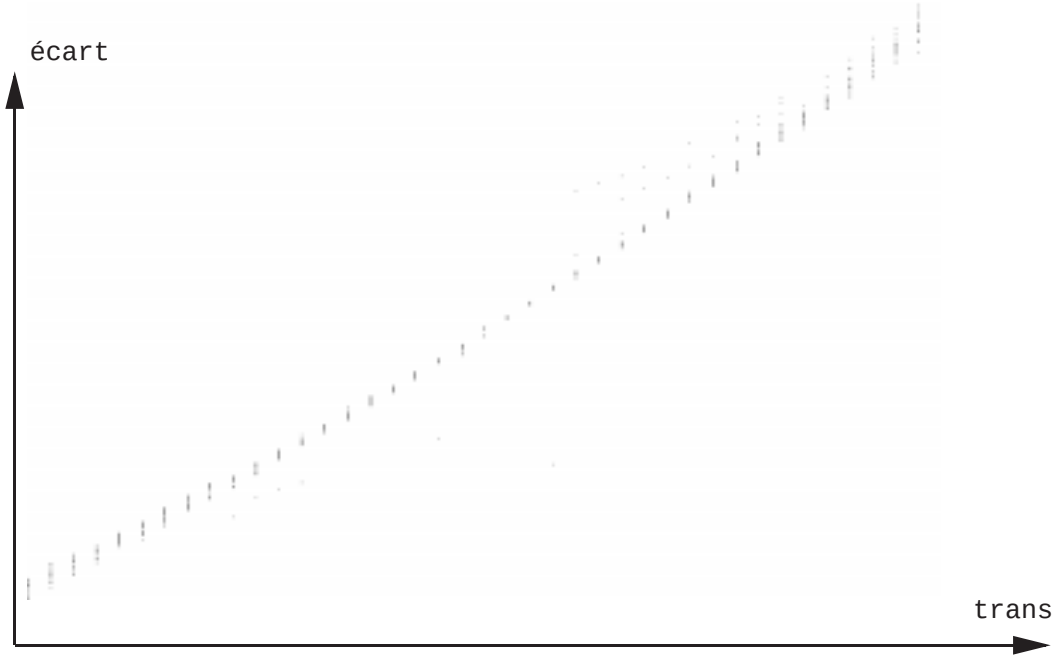


Figure 18. Les mesures (*trans, écart*) correspondant à celles de la figure 17 (note : les valeurs de `écart` sont communes aux deux figures).

L'interprétation de ce résultat est claire : le capteur proprioceptif est défectueux. L'erreur moyenne (quadratique) entre `trans` et `proprio` est en faite égale à 1.5 cm, ce qui est énorme pour une commande de bras de robot.

À présent, posez le problème à quelqu'un dans ces termes : “le robot est commandé en position par `trans`, et dispose d'un capteur proprioceptif `proprio` qui fournit après exécution d'une commande la vraie position atteinte ; or on remarque expérimentalement que l'écart moyen entre `trans` et `proprio` est de 1.5 cm ; qu'en concluez-vous ?”. La plupart vous répondront que la commande est imprécise, puisque la vraie position atteinte n'est pas celle commandée.

Or, la seule comparaison de `trans` et `proprio` ne permet pas en fait de trancher sur l'*origine* du problème : est-ce la consigne qui est mal respectée, le capteur proprioceptif fiable ayant permis de repérer l'erreur ? ou est-ce au contraire ce dernier qui est douteux, la commande étant correcte ? ou est-ce une origine mixte, toutes les variables étant très imprécises ?

On retrouve, dans la confiance a-priori accordée à `proprio`, l'influence des représentations prédéfinies devant un problème posé de façon abstraite (nous-mêmes avions toujours envisagé les choses de cette façon, jusqu'au jour où nous nous sommes trouvés devant la figure 17). C'est pour cette raison, pensons-nous, que le capteur proprioceptif avait déjà été largement utilisé dans d'autres manipulations plus classiques (méthodologie top-down), sans que le problème ait jamais été remarqué (sans doute les résultats de certaines de ces manipulations ont-ils été affectés, mais les mauvais résultats ne pouvaient qu'être constatés sans qu'il soit possible d'en identifier les causes).

À ce point de vue, la démarche proposée, fondée sur les DEP, a eu l'avantage de concerner une troisième source sensorielle `écart`, qui permet de conclure (au vu de

la figure 18) : `proprio` est douteux[3] et `trans` est correct. Notons que ce diagnostic n'a pas nécessité de valider `écart` au préalable comme bien lié au déplacement effectué par la caméra. En fait, un tel diagnostic n'est même pas formalisable, il s'agit réellement du résultat d'une interprétation de notre part.

Nous reviendrons au §C.5 sur la leçon méthodologique à tirer de cet épisode.

B.3. L'expérience du Khépéra photophile

L'expérience du Khépéra photophile exploite l'histoire du KitBorg citée en introduction, et illustre un aspect de l'incrémentalité absent des expériences précédemment décrites.

L'idée est de reprendre l'expérience du chapitre 4 (apprentissage d'un comportement d'évitement d'obstacles), mais à partir d'un comportement initial non plus téléguidé mais programmé par un réflexe dirigeant le robot vers la lumière (comportement dit “photophile”). De ce comportement photophile émerge un comportement d'évitement d'obstacle, le robot contournant en fait les ombres projetées par ces obstacles (voir chapitre 1). Cet aspect du comportement est capturé par une DEP qui permet alors de reproduire un comportement d'évitement d'obstacle même dans un contexte non photophile.

L'identification d'une DEP apparaît ici comme une façon de *caractériser explicitement un comportement à l'origine seulement émergent.* On peut ainsi caractériser et intégrer un aspect “imprévu” du comportement du robot, pourvu que l'on puisse imaginer (de façon opportuniste) une DEP qui le caractérise.

L'expérience du Khépéra photophile s'est en fait déroulée en deux temps, le comportement photophile utilisé pour l'apprentissage de l'évitement d'obstacle ayant lui-même été appris à partir d'un comportement télécommandé :

- Dans un premier temps, nous avons identifié une DEP_p (pour “photophile”) lors d'un comportement téléopéré, dans un environnement non encombré, qui consistait à orienter le robot vers la lumière en agissant sur la vitesse de rotation `vrot` (la vitesse de translation étant constante, voir chapitre 4 §E). La DEP_p mettait pour cela en dépendance cette variable motrice `vrot` et un prétraitement `maxlum` calculé comme la cellule photoélectrique la plus fortement illuminée (les huit capteurs infrarouges étant pour cela utilisés de façon passive, i.e. comme des capteurs de lumière et non des proximètres). Notons que pour effectuer les comparaisons, les capteurs n'étaient pas étalonnés les uns par rapport aux autre.
- Dans un deuxième temps, nous avons utilisé la DEP_p pour reconstituer un comportement photophile : lisant une valeur de `maxlum`, la valeur inférée `vrot'` était utilisée comme commande motrice. Ce comportement reconstitué était exécuté dans un environnement à présent encombré d'obstacles. Ces obstacles étaient des blocs de hauteurs comparables, la hauteur de la source de lumière étant telle que les ombres projetées soient d'une taille adéquate pour voir se répéter le phénomène constaté dans l'histoire du KitBorg : le robot évitait naturellement les

---

[3] La raison a été trouvée plus tard : il s'agit d'un bug logiciel dans le système de contrôle du robot, qui rend aléatoire les bits de poids faibles lus sur les codeurs optiques des articulations. Imaginons toutefois que l'altération de la valeur lue ait été corrélée à un certain comportement (par exemple au temps d'exécution du mouvement), alors la comparaison entre `trans` et `proprio` aurait pu être exploitée a-posteriori d'une façon ou d'une autre. Dans le cas présent, nous doutons que l'imprévu que constitue ce bug puisse être réellement exploité, puisqu'il est aléatoire.

obstacles en cherchant à éviter leurs ombres. Cet aspect “émergent” du comportement a donné lieu à l'apprentissage d'une nouvelle DEP, notée DEP_z.

La DEP_p(`vrot maxlum`) est caractérisée par une dépendance gaussienne entre `vrot` et `maxlum`, et la DEP_z(`vrot dir prox`) est exactement la même que celle décrite au chapitre 4 (seules les données expérimentales changent puisqu'elles proviennent d'un autre comportement).

Le résultat de cette expérience est très satisfaisant : si on commande le robot par la variable `vrot'` issue de la DEP_z, il évite très correctement les obstacles, mais avec un “style” différent de celui obtenu au chapitre 4 à partir d'un comportement téléopéré. Notamment, en l'absence d'obstacle la trajectoire n'est plus rectiligne mais semble aléatoire.

C. PARTICULARITÉS DU CYCLE INCRÉMENTAL

Nous pouvons à présent, en nous appuyant sur les expériences ci-dessus, détailler quelques aspects de la démarche incrémentale que nous avons proposée au début du chapitre, c'est-à-dire rappelons-le :

- Programmer le robot pour effectuer un certain comportement dans un certain environnement, ce qui définit un certain contexte.
- Identifier une ou plusieurs DEP dans le contexte ainsi défini.
- Interpréter ces DEP selon le contexte, pour les exploiter (programmer le robot) via des variables virtuelles.
- À partir de ll'expérience que le concepteur a acquise à l'occasion des expérimentations, imaginer d'autres DEP pour développer ainsi le répertoire de DEP du robot.

Nous allons approfondir les aspects suivants, présents dans toutes les expériences décrites :

- Une DEP est exploitée après une opération d'interprétation de la part du concepteur, et cette interprétation dépend du contexte. (§C.1).
- L'apprentissage se fait lors de comportements bien définis, appelés “comportements professeurs” (§C.2).
- Nous apportons une réponse au problème de l'imprévu, en ce sens que si un imprévu est caractérisable par une DEP, alors la démarche incrémentale fournira cette DEP comme le moyen de tirer parti de cet imprévu (§C.3).
- L'incrémentalité brouille la vision claire que nous avions de la notion d'incertitude : le processus de réinterprétation d'une DEP étant de nature qualitatif, on ne voit guère ce que deviennent dans ce processus les incertitudes numériques liées à cette DEP (§C.4).
- Nous avons utilisé lors de nos expériences certaines variables artificielles (`place`, `couleur`, `distance`) qui sont de caractère prédéfini. Leur rôle au sein de notre démarche est très particulier, et leur utilisation demande des précautions méthodologiques (§C.5).

C.1. Les DEP doivent systématiquement être interprétées

Les expériences de ce chapitre visaient à montrer que les DEP, apprises dans un certain contexte, restent potentiellement exploitables par le concepteur dans d'autres contextes, via la notion de variable virtuelle. Pour cela, nous devons toujours plus ou moins “bricoler” un peu, changer les relations des DEP entre elles, combiner intelligemment les résultats obtenus afin d'en tirer des actions pertinentes.

“Trouver des utilisations nouvelles à une DEP déjà apprise”, nous appelons cela *réinterpréter* cette DEP au sein de nouveaux contextes. Par cette opération la DEP a en quelque sorte “pris un nouveau sens” pour le concepteur.

Une DEP mettant en oeuvre des variables motrices a une interprétation “canonique”, qui est d'inférer les commandes du robot d'après les valeurs sensorielles lues sur les capteurs. Ce faisant, on espère reconstituer un aspect du comportement ayant donné naissance à la DEP initialement.

Une autre utilisation standard peut être de reconnaître passivement que les données sensori-motrices sont compatibles avec telle ou telle DEP, et par cela de repérer certains aspects du comportement. Par exemple, lors d'un comportement photophile Khépéra peut “se rendre compte” qu'il est aussi en train d'éviter les obstacles (s'il a déjà appris par ailleurs une DEP interprétable comme représentant un évitement d'obstacles).

C.2. Comportements professeurs

Quelques travaux sur l'apprentissage ont adopté la démarche de “l'apprentissage par l'action” (learning by doing). Mel (1990) caractérise cette démarche ainsi :

> «A major goal of [Mel's] work has been to explore the limits of a learning procedure called *learning by doing,* that has thus far been relatively little exploited in the field of machine learning. Learning by doing … is described here in the context of sensory-motor behavior. In its basic form, the learning agent
> • begins with a repertoire of actions and some forms of sensory inputs,
> • exercises its repertoire of actions, thereby building a mental model that relates actions to their sensory consequences and vice-versa, and
> • runs its mental model, in one or both directions, to “envision” solutions to problems posed by the environment.»

Cette démarche vise souvent à apprendre des relations toujours valables, indépendantes du comportement du robot (c'est par exemple le cas du robot de Mel, MURPHY). Certaines relations, cependant, ne peuvent être observées que lors de comportements bien spécifiques.

Nous appelons “comportement professeur” tout comportement permettant l'observation d'une dépendance utile entre les variables du robot, qui peut alors donner lieu une DEP. Un comportement professeur n'est pas forcément utile en soi, à la limite son seul intérêt peut être de permettre l'apprentissage d'une DEP.

Nous avons rencontré de nombreux comportements professeurs dans ce document, par exemple :

- Au chapitre 4, dans l'expérience de la bassine lumineuse, les protocoles d'exploration systématique ou aléatoire avaient le rôle de comportements professeurs. L'importance du protocole expérimental sur les résultats, que nous

avions constaté à cette occasion, devient un exemple de l'importance du comportement professeur sur l'apprentissage.

- Toujours au chapitre 4, le comportement téléguidé du Khépéra était un comportement professeur, qui met le robot en situation d'apprendre à éviter les obstacles.
- Dans ce chapitre-ci, le comportement photophile du Khépéra au §B.3 était un autre comportement professeur pour l'évitement d'obstacle.
- Dans l'expérience de la mire, le premier comportement (faire des mouvements aléatoires *en partant d'une distance fixe)* permettait l'apprentissage de la première DEP(`écart trans`). Un comportement totalement aléatoire, par exemple, n'aurait pas assez contraint l'interaction sensori-motrice pour permettre d'apprendre quelque chose d'utile.
- Les mouvements systématiques qu'utilise Mel (1990), destinés à parcourir l'espace des configurations possibles du robot, définissent un comportement professeur.

Une idée que nous n'avons pas pu expérimenter mais qui nous semble parlante est celle de voir *l'imitation* comme un comportement professeur. L'idée est que l'imitation de la forme *extérieure* des mouvements d'un “professeur” est de nature à faire apprendre certaines relations *propres* qui dirigent ces mouvements, de façon à être capables plus tard *non plus de les imiter mais de les reproduire.*[4]

Parmi les exemples de comportements professeurs ci-dessus, nous avons cité la télécommande par un opérateur. Dans le cadre habituel de l'apprentissage par l'action, ce type de procédé serait instinctivement assimilé à un apprentissage supervisé du même type que la donnée directe d'une base d'exemples par le concepteur. Cependant, nous ne voyons pas de différence de principe, dans le rôle de “professeur”, entre un comportement téléguidé et un comportement résultant d'une imitation ou d'un réflexe :

L'apprentissage que rend possible un comportement professeur ne dépend pas de la façon dont il a été réalisé, mais de son résultat effectif, c'est-à-dire du contexte dans lequel le robot se trouve placé de par ses actions dans son environnement — et ce quelle que soit l'origine de ces actions.

C.3. La gestion de l'imprévu

Notre démarche exploite l'expérience passée du robot par l'utilisation de variables virtuelles inférées via des DEP. Il s'ensuit que *les imprévus dont on peut tirer parti sont ceux qui peuvent donner naissance à une DEP.*

Par exemple, l'histoire du KitBorg mettait en jeu un robot ne disposant pas de capteurs de proximité (sonars, proximètres). Un aspect imprévu de son

---

[4] Il est clair que l'efficacité d'un tel apprentissage par imitation dépendrait fortement de la richesse des mouvements à imiter par rapport aux facultés d'apprentissage du robot. Pour faire une analogie avec l'expérience humaine, nous pouvons songer aux arts martiaux. La pédagogie des arts martiaux est fondée sur l'imitation d'un professeur, et il est caractéristique que la perception même que l'élève a des mouvements du professeur évolue avec l'expérience. Au début, il ne va en saisir que certains aspects, et plus il parvient à “internaliser” la caractérisation de ces mouvements, plus il est capable de voir les détails et de comprendre leur importance. Il s'agit en un sens d'un processus incrémental.

comportement, comme le fait d'éviter les obstacles, ne pouvait être que constaté par un observateur humain, mais ne permettait pas d'enrichir la “connaissance” du robot. Si au contraire le robot dispose de capteurs adéquats, l'aspect imprévu peut être reconnu ou appris, et donc exploité ; c'est ce qu'apporte l'expérience du Khépéra photophile par rapport à l'histoire du KitBorg.

De même, pour tenir compte du syndrome des pas de porte en tant que phénomène imprévu, il est nécessaire de pouvoir caractériser un glissement par ses conséquences sous forme de DEP. Un robot muni d'une boussole pourrait repérer ce glissement, par exemple, en constatant la rupture d'une dépendance entre sa commande en rotation et la variation de l'azimut suivi. Chez un robot non muni d'un tel dispositif (ou équivalent), notre démarche incrémentale ne serait pas en mesure de caractériser et exploiter cet imprévu.

C.4. Incertitude et interprétation

Revenons à présent sur le problème déjà signalé au sujet du “bricolage” de la figure 12. Une DEP a été identifiée lors d'une expérience concrète dans un certain contexte. Cette DEP est de nature probabiliste, et les distributions obtenues représentent les incertitudes déterminées expérimentalement dans ce contexte. Lorsque nous changeons de contexte, nous voulons réutiliser cette DEP en en réinterprétant les termes par rapport au nouveau contexte. Dans cette opération, il est sûr que la structure de la DEP doit rester la même (i.e. son aspect qualitatif, qui fonde son interprétation), mais pourquoi insistons-nous pour dans le nouveau contexte maintenir aussi la caractérisation *quantitative* de l'incertitude ?

La réponse est que d'une part nous n'avons pas le choix, et que d'autre part *l'aspect quantitatif a une influence essentielle sur l'interprétation.*

Nous n'avons pas le choix parce que si nous voulons n'utiliser, pour gérer l'incertitude, que des paramètres identifiés expérimentalement et non fournis a-priori, les seuls disponibles sont ceux qui ont été déterminés à l'origine. Cet argument serait spécieux si les valeurs de ces paramètres étaient totalement arbitraires par rapport au nouveau contexte, mais ils ne le sont pas *parce que nous connaissons le contexte concret qui leur a donné naissance.* Prenons un exemple pour illustrer cela.

Nous avons mené une expérience dans laquelle nous tentions de combiner formellement deux DEP apprises indépendemment : celle décrite au chapitre 4 pour l'expérience du Khépéra téléguidé, et celle apprise lors d'un comportement photophile (décrite dans ce chapitre au §B.3). L'idée était de commencer à formaliser certaines réutilisation des DEP dans des cas très simples (en toute généralité, dans notre approche actuelle, l'exploitation des DEP est entièrement laissée à la discrétion du programmeur).

Les résultats ont été très décevants. Nous avons alors réalisé que concernant l'évitement d'obstacle, le robot avait appris de façon très prononcée à aller tout droit en l'absence d'obstacle, ceci parce que les opérateurs ayant téléguidé les comportements professeurs avaient tous par défaut adopté cette stratégie. Cette tendance s'est avérée nettement supérieure à celle de la DEP photophile sur le comportement combiné, et en définitive le robot allait tout droit en l'absence d'obstacles en ignorant la lumière, alors que nous aurions voulu au contraire voir dans ce cas l'aspect photophile dominer.

L'interprétation de la DEP en tant qu'implémentation possible d'un comportement d'évitement était en fait trompeuse. L'évitement n'est réellement défini qu'en présence d'obstacles. Il aurait fallu, pour mieux justifier cette interprétation, que les variables motrices virtuelles de la DEP ait été indéfinies ou très incertaines dans le cas d'un espace libre, afin de ne pas contraindre le comportement. Ceci, nous l'avons compris en regardant les paramètres des gaussiennes composant la DEP, et nous avons pu réagir pour corriger ce défaut.

On voit donc en quoi les résultats quantitatifs sur les incertitudes peuvent changer l'interprétation que l'on accorde aux variables virtuelles. Plutôt que de conclure que les incertitudes apprises dans un contexte ne sont pas pertinentes dans un autre, ce qui n'est guère constructif, il faut, là encore, *exploiter les informations sur ces incertitudes que fournit la connaissance du contexte concret les ayant déterminées.* Le problème, là encore, est un problème d'interprétation.

C.5. La gestion des variables artificielles

Nous avons dans nos exemple accepté l'emploi de variables artificielles. Mais il est clair que dès que l'on souhaite un fonctionnement *automatique* d'un robot, c'est-à-dire sans intervention humaine, les variables artificielles ne peuvent plus être admises.

Première idée : recourir à un modèle en termes prédéfinis

Une première possibilité pour remplacer les variables artificielles est la suivante. Nous avons défini ces variables comme “fournies au clavier” par le concepteur. Cependant, en pratique les variables de ce type sont rarement fournies par le concepteur, mais plus souvent calculées par des modèles. Nous pourrions donc recourir à ce procédé pour continuer à utiliser de façon automatisée des variables telles que `couleur` ou `distance`.

Cette solution n'est pas acceptable pour nous, car les variables concernées sont toujours des termes “prédéfinis”, et nous avons vu au chapitre 2 que la détermination automatique de tels termes pose le problème des conditions de validité. La méthode consistant à fournir les valeurs au clavier était destinée à contourner ce problème : le concepteur assiste à l'expérience, il ne s'agit pas d'un système informatique mais d'un être intelligent, et on lui fait confiance pour fournir les valeurs correctes quand on les lui demande.

Deuxième idée : recourir à des variables virtuelles

Lorsqu'on désire un fonctionnement automatique du robot, nous proposons de transformer les variables artificielles non pas en termes prédéfinis mais en variables virtuelles, via l'apprentissage d'une DEP.

C'est exactement ce que nous avons fait lors de nos expériences. À l'origine `place` et `couleur` par exemple sont des variables prédéfinies (position et couleur de la bande de papier) et donc artificielles (fournies par le programmeur). Nous avons mis ces variables en dépendance empirique avec d'autres, et nous les avons ensuite utilisées uniquement via les inférences `place'` et `couleur'` tirées de ces dépendances. Une variable artificielle, définie par sa fonction, se transforme par ce moyen en une variable virtuelle, définie par son mécanisme d'inférence.

Plus précisément, la méthode pour transformer une variable artificielle V (devant être fournie par l'opérateur) en une variable virtuelle V' (pouvant être calculée automatiquement par le robot) est la suivante :

- On cherche avec quelles autres variables sensori-motrices ou virtuelles cette variable V pourrait donner lieu à une DEP. Par exemple, soit A un prétraitement que l'on pense pouvoir relier à V.
- On identifie la DEP(AV) *lors d'une expérience effective.*
- On remplace V par V', variable virtuelle inférée via la DEP après lecture de A.

Les variables artificielles ne servent en pratique qu'à superviser une phase d'apprentissage. Si l'on désire que l'apprentissage lui-même se passe de supervision directe, les variables artificielles sont même complètement exclues (notons qu'une forme de supervision reste possible via le comportement professeur, qui peut être par exemple télécommandé).

Conclusion : une position méthodologique

En conclusion de ces remarques, nous adopterons plus généralement la méthodologie suivante, qui complète la démarche incrémentale présentée dans ce chapitre :

Avant d'utiliser une variable quelconque dans un processus automatique, il faut impérativement :

- *connaître le mécanisme effectif qui permettra de calculer cette variable*
- *tenir compte de cette connaissance pour adapter l'utilisation effective qui sera faite de cette variable par le programmeur.*

Les variables sensori-motrices, virtuelles et artificielles (fournies au clavier) satisfont ces deux conditions. Les variables provenant de modèles classiques ne satisfont pas la seconde : elles sont le plus souvent utilisées en fonction de leur sens prédéfini, et non du mécanisme effectif utilisé pour les calculer.

Ceci peut être bien illustré en revenant sur la commande en position du bras de robot. Supposons que l'on doive modéliser la possibilité d'erreurs de positionnement de la pince (par exemple pour une application de manipulation fine). Le raisonnement standard serait le suivant. On cherche la position `posvraie` effective de la pince dans la scène, par rapport à la base du robot. On ne connaît comme estimation de cette position que `proprio`, le capteur proprioceptif du contrôleur du robot. La précision de ce capteur dépend d'une part de celle des pas codeurs servant à asservir les moteurs, mais également de certains phénomènes mécaniques de frottement et d'inertie, qui font que certaines commandes faibles (de l'ordre de un ou deux millimètres) provoquent une vibration du robot mais ne semblent pas réellement être effectués. Pour tenir compte de ces incertitudes, on introduit alors une erreur et on écrit `proprio` = `posvraie`$+\varepsilon$, ε étant une variable aléatoire gaussienne de moyenne 0 et de variance σ (dans notre cadre, nous écririons p (`proprio` | `posvraie`) = G (`proprio` | `posvraie`, σ)). σ serait alors

considérée petite par rapport à la valeur de `posvraie` (par exemple pour permettre des approximations au premier ordre dans les calculs).

Le problème de ce paramètre σ est qu'il n'est pas identifiable expérimentalement, seulement estimable a-priori. En effet, pour l'identifier, il serait nécessaire de mesurer expérimentalement un ensemble de couples (*posvraie, proprio*), or `posvraie` n'est pas supposée systématiquement mesurable — raison d'ailleurs pour laquelle on cherche à l'estimer via `proprio`…

De plus, nous avons vu que le capteur proprioceptif est défectueux, et qu'en pratique σ est égal à 1.5 cm, ce que personne n'aurait imaginé a-priori (les approximations assumant σ faible se seraient donc révélées inadéquates). Rappelons que ce défaut a été détecté via une DEP entre `proprio` et `écart`, et non pas entre `proprio` et `posvraie`. Ce n'est que notre interprétation des phénomènes observés qui a permis d'expliquer ce défaut en termes prédéfinis (c'est-à-dire dans les termes “le capteur proprioceptif est défectueux” — défectueux faisant référence à une utilisation prédéfinie).

Dans ces conditions, nous proposons donc de ne pas utiliser de variables telles que `posvraie`, mais uniquement des variables mesurables expérimentalement, soit par le robot lui-même (variables sensori-motrices), soit par l'expérimentateur (variables artificielles lues au clavier), soit par une opération d'inférence appelant une réinterprétation ultérieure (variable virtuelle).

CHAPITRE 6

LA REPRÉSENTATION CONTINGENTE

If you have a hammer, don't treat
everything as if it were a nail.
Auteur inconnu

Nous proposons dans ce court chapitre une notion de représentation, dite “représentation contingente”, alternative à la représentation prédéfinie critiquée au chapitre 2. La section A motive et définit la notion de représentation contingente, la section B compare notre approche à deux autres approches, celles de Lloyd (1988) et de Agre (1988), et enfin la section C discute de ce que devient dans cette perspective inhabituelle la notion de “but” ou de “tâche”.

A. UNE NOTION DE REPRÉSENTATION NON PRÉDÉFINIE

Rappelons la motivation qui nous a conduits au chapitre 2 à nous intéresser à la notion de représentation par rapport à l'incrémentalité : «il faut préciser ce qui, dans la structure d'un robot, peut être amené à changer, et ce qui au contraire doit rester invariant pour ne pas perdre l'expérience passée que l'on souhaite continuer à exploiter. La notion de représentation est pour nous le moyen de faire cette distinction, de désigner ce qui doit rester constant et pour quelle raison».

Dans l'approche dominante de la “représentation prédéfinie”, l'invariant d'une représentation est sa *fonction* (comment elle permet de réaliser certaines tâches). La fonction caractérise et définit a-priori la représentation. Nous avons montré au chapitre 2 que cette approche n'est pas adéquate pour la robotique autonome, car en pratique elle repose sur la possibilité de modifier l'environnement pour l'adapter aux représentations définies *a-priori* pour le gérer.

Nous proposons de remplacer la représentation prédéfinie par une notion de représentation *contingente,* le contingent étant ici le contraire du prédéfini[1] :

> *Principe de la représentation contingente* : Une représentation est une certaine structure telle que :
> (1) Les contextes dans lesquels elle est interprétable ne peuvent pas être limités a-priori,
> (2) Une interprétation est toujours contingente au contexte.

[1] Le terme de contingence a en français plusieurs sens contradictoires. “Contingent” n'est surtout pas à prendre ici au sens de “sans importance”, ni au sens de “aléatoire”. Il faut le comprendre au sens philosophique de “qui peut ou non arriver”. Un événement contingent peut prendre une importance pour des raisons historiques, dans un contexte particulier, mais n'est ni déterminé ni nécessité par l'histoire ou le contexte — il y survient, et seulement alors il peut devoir ou non être pris en compte. Le contingent est donc bien le contraire du prédéfini. Cette notion est bien développée dans les livres de Stephen Gould (1991, 1993), dans le domaine de l'évolution des espèces.

On retrouve dans cette définition une variante affaiblie de la propriété fondamentale des représentations classiques : le fait d'être “systématiquement interprétable” (Fodor & Pylyshyn 1989, Harnad 1990). Le sens littéral de ce terme serait : “peut systématiquement subir une interprétation”. Cependant, l'acception classique exige de plus que cette interprétation soit indépendante du contexte (Fodor & Pylyshyn 1989), ce que nie notre propriété (2). La propriété (1) est alors à comprendre comme un retour au sens littéral de “systématiquement interprétable”, affaibli du fait que nous admettons, dans notre formulation, que dans certains contextes une représentation puisse ne pas être interprétable (l'important étant l'impossibilité de caractériser a-priori tous ceux où elle l'*est).*

Le caractère invariant d'une représentation n'est donc pas sa fonction mais bien sa structure (les mécanismes qui la définissent).[2] L'interprétation est alors le processus par lequel on adapte une représentation à un contexte donné. Notons de plus que puisque cette interprétation est par principe toujours contingente au contexte, *elle ne peut être faite “qu'en situation” et non a-priori* (il faut se trouver devant un problème concret pour pouvoir interpréter ce que signifie une représentation).

Le changement d'optique que tente d'esquisser cette approche de la représentation peut être illustrée sur la figure 2, opposée à la figure 1 correspondant au contraire à la représentation prédéfinie.

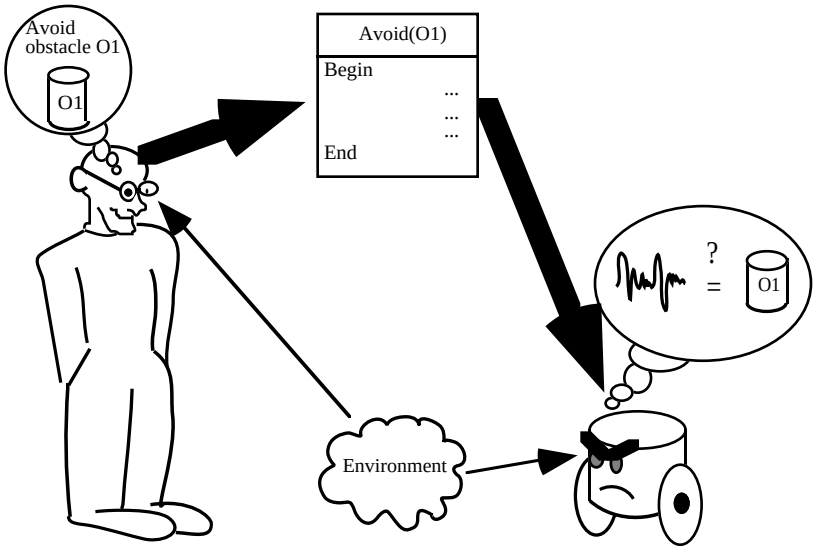


Figure 1. Illustration de la représentation prédéfinie. L'interprétation d'une représentation (bulle du personnage) est fixée, indépendante du contexte, de nature abstraite. Cette approche est confortable pour le concepteur, mais en robotique autonome il est difficile de définir une relation formelle entre ces termes abstraits et les variables sensori-motrices effectivement mesurables (bulle du robot).

---

[2] En psychologie, on constate que c'est la fonction des organismes vivant que ceux-ci cherchent à maintenir face à des lésions (cf Merleau-Ponty 1942). Il est alors normal de vouloir caractériser des organismes par leur fonction. Dans notre optique, de roboticien, les invariances fonctionnelles ne sont pas une donnée fondamentale mais plutôt ce que nous aimerions *expliquer* en termes structurels : *comment* une même fonction peut-elle être réalisée par des mécanismes différents, *comment* le passage opportuniste (voire improvisé) des uns aux autres est-elle possible ?

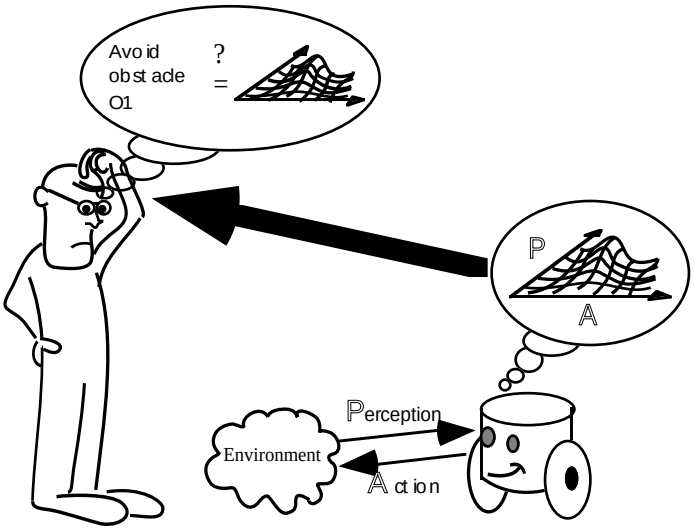


Figure 2. Illustration de la représentation contingente. Une représentation s'exprime en termes propres au robot, sans référence à une fonction définie par rapport à un monde extérieur (bulle du robot). Pour l'exploiter, le concepteur doit d'abord interpréter, en fonction du contexte, ce qu'elle représente (bulle du personnage). Dans ce dessin, c'est une DEP (cf chapitre 5) qui joue le rôle de la représentation.

Pour finir, notons que la notion de représentation contingente témoigne d'une volonté d'adopter une “position de principe” forte sur la nature de la représentation, mais ne constitue en aucun cas une tentative d'établir une “théorie” de la représentation. En effet, le principe proposé fait appel à la notion *intuitive* d'interprétation, que nous considérons comme non formalisable ; ce n'est donc pas un principe formel et nous ne cherchons pas à le rendre formel. En particulier, notons que *le fait qu'une structure donnée satisfasse ou non à ce principe ne peut pas être démontré.*

Les DEP introduites au chapitre précédents sont candidates pour être considérées comme des représentations au sens ci-dessus. Nous avons montré au chapitre précédent qu'une DEP est potentiellement interprétable dans plusieurs contextes et que son interprétation (la façon dont nous l'utilisons) est variable. Pour faire effectivement passer les DEP au rang de représentations, il faut considérer qu'elles sont potentiellement interprétables dans un nombre illimité de contextes, mais comme nous venons de dire cela ne peut pas être prouvé et ne saurait constituer qu'une hypothèse de travail.

Les DEP ne sont sans doute pas les seules formes de représentations internes envisageables. Nous espérons bien trouver plus tard d'autres structures plus riches dans ce rôle.[3]

---

[3] Une faiblesse essentielle des DEP en tant que représentations internes est leur incapacité à représenter les aspects temporels du comportement. Nous pensons que cela nous restreint à des comportements de type “réflexes”. Ceci n'est toutefois qu'une pure spéculation ; et d'autre part l'étude des systèmes purement réflexes semble loin d'avoir atteint ses limites (autre spéculation)…

B. COMPARAISON À D'AUTRES APPROCHES DE LA REPRÉSENTATION

Les travaux proposant des approches nouvelles de la notion de représentation sont peu nombreux. Il s'agit plus souvent de défendre ou de critiquer la notion standard de représentation prédéfinie. Nous allons ici comparer notre approche à deux autres dont les préoccupations ont des points communs avec les notres : celles de Lloyd (1988) et de Agre (1988).

B.1. Comparaison avec l'approche de Lloyd (1988)

L'idée de fonder la représentation sur des dépendances entre valeurs sensorielles n'est pas neuve, on la trouve notamment chez Lloyd (1988), et même chez un objectiviste comme Marr (1982). Notre originalité par rapport à ces auteurs est de considérer les aspects ainsi caractérisés comme variables selon le contexte. Nous détaillons ici notre différence avec Lloyd, dont certaines préoccupations se rapprochent beaucoup des nôtres, mais dont la volonté de formaliser la notion d'interprétation rend en fin de compte l'approche incompatible avec la nôtre.

Lloyd a élaboré une théorie de la représentation en se référant explicitement à la robotique comme cadre principal de réflexion, et il fonde son approche sur une structure d'inspiration similaire à nos DEP.

Voici ce qu'il propose (réexprimé schématiquement à notre manière) :

Étant donné un ensemble “d'événements”, Lloyd affirme qu'un événement R est une représentation s'il satisfait à trois conditions C1, C2 et C3 :

C1: R dépend d'une conjonction de plusieurs événements d'un sous-ensemble donné. Par exemple, R est le changement d'état d'une porte AND, ou l'activation d'un neurone formel quand la majorité de ses sysnapses sont actives.

C2: R vérifie un certain critère de “convergence”, détaillé ci-après.

C3: R a la capacité de causer soit une modification directe du comportement du robot, soit l'apparition d'une autre représentation.

Présentons le critère C2 sur un exemple simple. Soient deux cellules photoéléctriques, de sorties supposées booléennes pour simplifier. Les événements considérés sont les activations des cellules. Ces cellules peuvent être activées conjointement, ce qui constitue un événement R. Dans quelles situations R peut-il être appelé une représentation ?

Lloyd compare alors les trois situations de la figures 3.

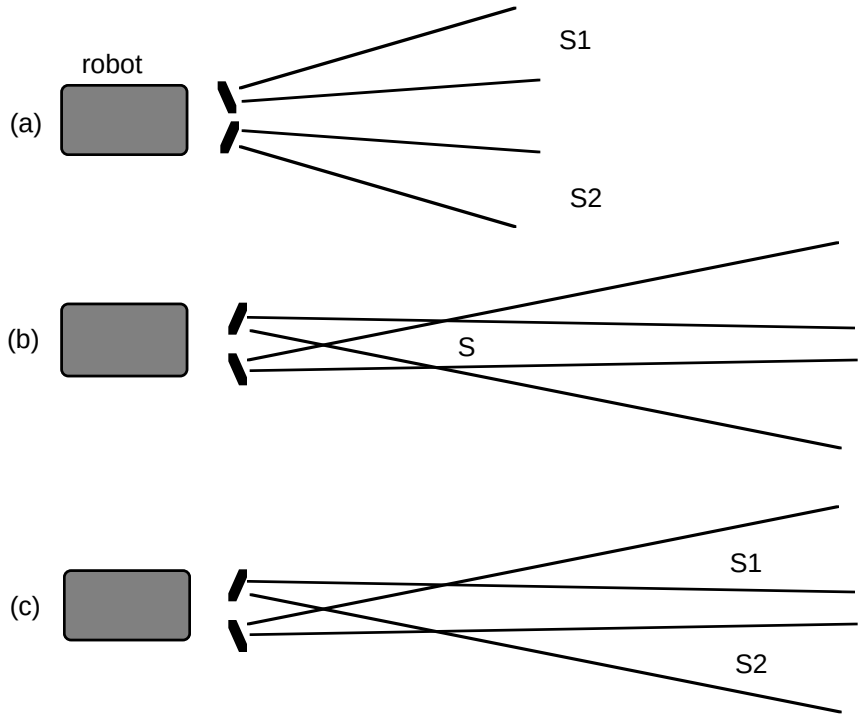


Figure 3. Un robot est muni de deux cellules photoélectriques directives, dont les cônes schématisent les champs récepteurs. Dans quelles conditions les deux cellules sont elles activées conjointement (on les suppose booléennes pour simplifier) ? (a) Les champs récépteurs sont disjoints, la conjonction provient donc de deux sources séparées S1 et S2. (b) Les champs ont une intersection, et la conjonction peut provenir d'une source unique S. (c) Même dans le cas précédent, toutefois, la conjonction peut encore provenir de sources séparées en S1 et S2.

Dans la situation (a), les cellules sont activées en même temps mais la cause est un stimulus dit “composé”, provenant de sources disparates (S1 et S2 sur la figure). R ne peut pas être qualifié de représentation dans ce cas, à cause de l'impossibilité de considérer “ce qui est représenté” comme un objet isolable spatialement dans l'environnement et sur lequel R permette de se focaliser.

Dans la situation (b), par contre, il y a possibilité de se focaliser du fait de la convergence des champs réceptifs des deux cellules : R est expliquable comme provenant d'une source unique S, spécifiable isolément du reste de l'environnement. R peut être alors appelé une “représentation”, pour peu que le comportement du robot soit influencé par sa présence ou son absence (critère C3).

Certes R ne permet pas de distinguer cette situation (b) de la situation (c) (stimulus composé), mais il suffit qu'une explication de type “stimulus simple” *existe* pour qualifier R de représentation. Le contenu représentationnel (interprétation) de R est alors le stimulus simple *le plus probable* permettant d'expliquer la conjonction observée (une représentation peut donc se révéler fausse : dans la situation (c), c'est l'interprétation (b) qui serait assumée).

Le “critère de convergence” général C2 que propose Lloyd est alors le suivant : étant donné R une conjonction de plusieurs événements, R ne peut être une représentation que s'il existe un ensemble de stimuli “simples” (single mutually exclusive stimuli) dont dépend chacun des événements en conjonction. Le contenu

de R est alors le stimulus simple le plus probable sachant que la conjonction est réalisée.

Lloyd admet donc l'existence dans l'environnement de stimuli “simples”, et conditionne formellement sa définition d'une représentation au fait que son activation puisse être causée par de tels stimuli. Il est donc nécessaire de décrire préalablement l'environnement en termes d'événements simples pouvant être combinés, *c'est-à-dire de proposer un modèle prédéfini de l'environnement.* Par exemple, pour traiter le cas de la figure 3, il faudrait modéliser l'environnement comme une combinaison d'événements élémentaires “présence d'une source de lumière à la position (x,y), avec x et y réels”.[4]

Lloyd considère qu'il ne fait qu'une concession minime à la position objectiviste (il appelle sa position “modest realism”). Mais, concrètement, le jour où cette approche devra servir à construire un robot, il lui faudra effectivement choisir un modèle particulier du monde sur lequel reposera toute la validité de sa réflexion. Clairement, Lloyd utilise le robot comme métaphore, comme une abstraction, et se place de fait dans un monde-jouet.

Notre opinion est qu'à partir du moment où une certaine conjonction d'événements est observée, elle “représente potentiellement” toutes les raisons qui peuvent l'expliquer, mais ces raisons n'étant par essence pas formalisables une interprétation donnée ne peut par contre émerger que dans un contexte donné. C'est parcequ'il essaye de formaliser la notion d'interprétation comme “le choix d'un stimulus simple dans un ensemble préexistant” que nous tombons en désaccord avec Lloyd.

B.2. Comparaison avec la “deictic representation” de Agre (1988)

Nous avons dans l'introduction cité “Pengi” de Agre & Chapman (1987) comme un exemple de système fonctionnant dans un environnement simulé (jeu vidéo) mais néanmoins “naturel” — au sens que la simulation n'a pas été conçue en fonction du système dont elle constitue l'environnement. Pengi illustre toute une réflexion sur la nature de l'action, détaillée dans la thèse de Agre (1988), et avec laquelle nous sommes en grande partie d'accord. Les grands thèmes concernent principalement la nature fondamentalement “routinière” de l'action d'un agent dans son environnement, et le fait que l'action ne doit pas être vue comme une résolution de problème mais comme la conduite, par une continuelle improvisation, d'une interaction avec l'environnement. Le monde est excessivement difficile à décrire, mais en un sens il est relativement facile d'y agir.

Cette réflexion a amené Agre à proposer une approche de la représentation dite “représentation déictique”. L'idée est qu'un agent dans le monde doit se représenter celui-ci non pas sous la forme de descriptions “externes”, sans référence à l'agent lui-même, mais sous la forme de relations opérationnelles à l'agent, par exemple *le-verre-que-je-tiens-à-la-main.* La représentation désignée sous ce nom est appelée une “entité” par Agre, et peut avoir plusieurs “aspects” modulant la relation, comme *le-verre-que-je-tiens-à-la-main-est-vide* ou *le-verre-que-je-tiens-à-la-main-est-fragile.*

Cette approche est intermédiaire entre une approche fonctionnelle et une approche structurelle de la représentation :

---

[4] On retrouve en cela l'influence de la modélisation ensembliste fondant la théorie standard des probabilités.

- Une représentation déictique n'est pas définie par sa fonction : *le-verre-que-je-tiens-à-la-main* n'est pas manipulé par un ensemble de règles prédéfini, et le concepteur a un effort d'interprétation à faire, faisant largement appel au sens commun, pour utiliser cette représentation au sein d'une tâche particulière (boire, remplir, casser, offrir…).
- Elle n'est pas définie non plus par sa structure, car l'expression en termes sensoriels de *le-verre-que-je-tiens-à-la-main* reste délicate à déterminer… cela reste un niveau d'abstraction difficile à “ancrer”.

La représentation déictique ne permet pas de définir des “représentations internes”, mais reste une notion tournée vers le concepteur, un guide de conception mettant bien le doigt sur les caractéristiques importantes de l'activité “en situation” qu'on aimerait obtenir chez un système artificiel.

Au contraire, l'objectif de la représentation contingente est justement de décrire des représentations *internes.* Une représentation doit s'exprimer dans les termes propres du robot.

Notre réflexion sur la représentation serait en fait plus proche de celle d'Agre sur la notion de “plan”. Agre & Chapman (1988) distinguent deux sortes de “plans” : ceux qui spécifient une suite d'instructions à effectuer, et ceux qui fournissent des indications concernant l'ordre des opérations à effectuer. Le premier est ce qu'on appelle “plan” en IA , le second ce qu'on appelle “plan” dans la vie de tous les jours, comme par exemple une recette de cuisine. La différence est que si un plan, dans la vie de tous les jours, est une ressource parmi d'autres, un certain type d'information destinée à un humain, un plan au sens de l'IA est censé totalement déterminer les actions d'une machine. C'est en définitive la même distinction que nous faisons à propos de la notion de représentation.

C. DES ROBOTS “POUR QUOI FAIRE” ?

Avec l'adoption d'une approche structurelle de la représentation, nous abandonnons le guide de conception pourtant difficilement contournable qu'est la notion de but (ou de tâche). Or, on développe toujours un robot “pour faire quelque chose”.

C.1. Les buts du robot : une notion émergente

Le problème du but est à notre avis *totalement* inclus dans celui de l'interprétation. Le but d'un robot est uniquement dans l'oeil de son concepteur, et nous ne souhaitons pas encore le considérer comme une notion fondatrice et systématique du comportement. Cette position est contraire aux approches habituelles de la robotique, dont les fondements sont souvent ces notions de but ou de tâche.

Pour détailler notre pensée, rappelons d'abord qu'au chapitre 2 nous avions plaidé pour la possibilité de décider des buts du robot de façon opportuniste, d'accorder plus d'importance à ce qu'un robot sait faire et à son potentiel de développement qu'à ce que le robot ne sait pas encore faire.

Cette démarche opportuniste revient à faire de la notion de but quelque chose d'émergent. Par exemple, l'évitement d'obstacle ne constituerait pas un but pouvant *expliquer* un mécanisme comportemental en ce sens que celui-ci aurait résolu celui-

là avec succès. Il ne semble pas nécessaire de faire intervenir a-priori l'orientation vers un but comme fondement même du mécanisme comportemental, mais plutôt comme une propriété justifiant l'existence de ce mécanisme (la métaphore biologique dirait “justifiant qu'il se soit maintenu par sélection naturelle”).

La notion de but est donc (pour nous) à considérer comme un résultat, et non comme une cause, du processus de développement.[5]

Pour nous inspirer (exceptionnellement!) d'une métaphore biologique, notons que même les éthologistes ne s'accordent pas sur les notions traditionnelles de but et de renforcement en tant que causes du développement de certains comportements. Gardner & Gardner (1991), par exemple, réinterprètent de façon très convaincante les phénomènes de conditionnement opérant étudiés dans les boîtes de Skinner. Ils montrent que les comportements observés ne semblent pas en fin de compte résulter de processus de renforcement, mais plutôt de processus “feedforward” de déclenchement de comportements “naturels” par des stimuli appropriés. Typiquement, le fait qu'un pigeon apprenne à frapper sur un bouton rouge pour obtenir à manger ne serait pas dû à un processus de renforcement, mais serait la manifestation d'un comportement pré-alimentaire (pre-feeding behaviour) qui se déclenche naturellement quand le pigeon a faim ou quand de la nourriture est évoquée par association. Cette interprétation semble largement étayée par de nombreuses contrexpériences.[6]

Nous avons également retrouvé des échos à ces idées dans un livre consacré à l'histoire des outils (Feller & Touret 19??). Il ne s'agit pas d'un ouvrage scientifique, mais le sujet reste pertinent, car entre les “outils” physiques des artisans et les “outils” conceptuels destinés à l'IA, la parenté reste forte. Voici ce qu'écrivent ces auteurs :

> «Il nous apparaît impossible de supposer un outil créé par un besoin d'objet particulier. L'idée qu'un objet, encore inexistant dans sa forme bien définie, puisse susciter une création d'outil est inacceptable. Un outil (non encore adapté à ce qu'on va lui demander) est toujours antérieur à ce destin ignoré. Il n'est pas exclu que le hasard intervienne dans la genèse d'un outil… C'est lorsque le produit de cet objet insolite, après beaucoup d'essais, est jugé utile, que l'outil pourra servir de façon systématique. … Si certains outils ont duré, c'est d'abord parce qu'ils avaient atteint la forme utilitaire la meilleure.»

> «L'outil subit le sort de tous les organes qui se sont modifiés pour une utilisation particulière : dans la spécialisation, il perd ses facultés d'adaptation, qu'il semble avoir épuisées dans son effort de transformation. L'outil très perfectionné, trop compliqué, est bien près de disparaître en cédant place à quelque machine. Il a perdu la possibilité de se transmuer en un outil meilleur. Il ne saurait être encore perfectionné, ou adapté à un métier autre que celui qui lui a donné sa forme, que par une redécouverte qui ramène souvent au prototype à partir duquel il s'est mis à évoluer.»

---

5 Cette position n'est ni définitive ni un jugement a-priori. Simplement, les approches classiques de la notion de but ne nous conviennent pas parce qu'elles semblent trop pauvres, et plus généralement nous pensons même que notre compréhension de cette notion est encore trop imparfaite pour en envisager véritablement une approche théorique. Peut-être qu'un jour nous comprendrons comment cette émergence pourrait être caractérisable à un plus haut niveau, et devenir alors réellement le fondement d'un certain mécanisme. Mais il s'agit d'un problème cognitif encore trop complexe, que nous ne sommes pas prêts à aborder dans le cadre de la robotique autonome, et nous préférons en attendant considérer le but comme une notion émergente.

6 Les mêmes auteurs citent même un exemple de dressage animal dans lequel le but du dressage (le numéro) avait été décidé a-posteriori après observation de certains comportement naturels qu'il suffisait de “bien habiller” pour en faire quelque chose d'impressionnant (il s'agissait de faire démarrer un juke-box par une poule, qui ensuite dansait sur la musique). Notre démarche de buts opportunistes va tout à fait dans ce sens.

C.2. Viabilité et autonomie : une perspective

Stewart (1994b) propose de remplacer la notion habituelle de but par une notion de “contrainte proscriptive” : ne pas faire ceci, ne pas faire cela (les buts s'exprimant plus habituellement par “faire ceci, faire cela”). Cette notion ne nous convient pas. Prenons l'exemple d'un robot devant éviter les collisions et ne pas rester immobile. D'une part, pour que ce robot fonctionne il faudra bien lui avoir dit *comment* réaliser ces buts proscriptifs, et le problème concret reste entier. D'autre part ces contraintes nous semblent artificielles : elles sont requises par le concepteur mais n'ont pas une pertinence propre au robot, intrinsèque à la façon même dont il fonctionne. Même si le robot heurte un obstacle et s'abîme, rien dans son fonctionnement propre n'en sera affecté, seule l'observation du concepteur ou une procédure spécifique de test pourront l'être.

Ce qui nous semblerait fondamental pour guider les processus de développement serait alors, plus qu'une notion de but, une notion de “viabilité” qui concernerait des systèmes *déjà autonomes par ailleurs.*

C'est là que la notion d'autonomie en tant que “clôture opérationnelle” (Varela 1989) devient très attrayante, et c'est dans une telle notion de viabilité que nous voyons l'enjeu du débat entre les approches fonctionnelles et structurelles de l'autonomie.

Varela (1989) a proposé la thèse suivante : tout système autonome est opérationnellement clos. Un système opérationnellement clos est *défini* par une organisation en un ensemble de processus dont le déroulement même dépend de leurs propres conséquences (“dépendant récursivement les uns des autres pour la génération et la réalisation des processus eux-mêmes, et constituant le système comme une unité reconnaissable dans l'espace (le domaine) où les processus existent”, Varela 1989 p.86). La clôture[7] opérationnelle définit un système non par une caractérisation extérieure, mais par les propriétés de ses mécanismes constitutifs.

Cela n'implique en rien qu'un tel système soit isolé, il peut avoir de riches interactions avec l'extérieur. Le problème de la viabilité[8] est alors de faire en sorte que la clôture opérationnelle ne soit pas brisée par ces interactions. Le changement d'état causé par les interactions doit permettre aux processus constituant le système de continuer à exister.

Cette approche nous semble essentielle à opérationnaliser et à creuser pour l'avenir de la robotique autonome.Cependant cela n'est pas encore fait ; en pratique la notion de viabilité définie ci-dessus n'est pas utilisable comme guide concret de développement, et nous devons encore nous appuyer sur la notion traditionnelle de but. C'est en définitive pour cette raison que nous tenons à laisser, via la notion d'interprétation, un rôle explicite au concepteur tant dans notre démarche incrémentale que dans notre définition de la représentation : la notion de but étant incontournable, nous en laissons la gestion au concepteur lui-même *afin de ne pas*

---

[7] La notion de clôture est ici la même qu'en mathématiques : un espace clos selon une certaine opération est un espace qui contient encore tout résultat de cette opération sur les éléments de l'espace.

[8] Le terme de viabilité peut être discutable ici, ne concernant pas des êtres vivants mais des systèmes autonomes. Nous employons ici viabilité au sens, dont nous pressentions plus haut la nécessité, d'une contrainte (sur les comportements du système) de nature plus fondamentale que la réalisation de buts.

fermer la porte à une future opérationnalisation de la notion structurelle d'autonomie. Si dans nos travaux nous n'avons pas cherché à mettre en pratique les idées de Varela concernant l'autonomie, nous considérons cependant comme essentiel le fait de rester *compatibles* avec ces idées.

CHAPITRE 7

CONCLUSION

Je n'ai pas compris et j'en fus impressionné. Je suis toujours impressionné par l'incompréhensible, car cela cache peut-être quelque chose qui nous est favorable.

Romain Gary, “Gros Câlin”

A. Synthèse

Nous avons dans cette thèse défini le sujet de la robotique autonome comme la construction de robots destinés à fonctionner en milieu physique et naturel. Les difficultés de ce sujet sont nombreuses et relativement neuves par rapport à d'autres domaines. Nous avons voulu les aborder sous un angle qui mette en valeur cette nouveauté et ne permette pas de l'éluder : nous avons donc choisi *la recherche d'une solution systématique au problème de l'imprévu* comme problématique particulière.

Cette problématique est extrèmement riche, et nous ne l'avons certes pas résolue. Nous avons essentiellement progressé dans cette voie en disant “par quel bout” l'aborder.

Tout d'abord, nous avons identifié un certain nombre de causes majeures empêchant de bien aborder les difficultés spécifiques que pose cette problématique de l'imprévu :

- *Une cause conceptuelle.* Les modélisations utilisées en robotique se fondent implicitement sur une notion de “représentation prédéfinie”. Leur validité repose sur l'existence d'un concepteur capable, pour trouver une solution à un problème, de rajouter des contraintes sur l'environnement du robot lorsque le problème posé ne contredit pas explicitement ces contraintes. Il s'agit en somme d'adapter l'environnement concret à la modélisation abstraite qui en est faite. Nous prétendons que cette approche n'est pas adéquate pour gérer des environnements naturels. La notion de représentation prédéfinie empêche de correctement formuler les problèmes spécifiques de la robotique autonome — et a-fortiori de les résoudre.
- *Une cause méthodologique.* Pour aborder un problème complexe, il faut parvenir à en isoler certains aspects qui peuvent faire l'objet de recherches spécifiques. Nous prétendons que la décomposition traditionnelle conception-adaptation n'est pas adéquate pour la robotique autonome, et ne permet pas, en particulier, d'aborder le problème de l'imprévu.
- *Une cause théorique.* L'intelligence artificielle est l'inspiratrice principale de la robotique autonome. La théorie fondatrice de l'IA, qui est celle de la logique formelle et des systèmes symboliques, est largement adoptée dans ce cadre. Nous prétendons qu'il manque à cette théorie le moyen de gérer explicitement la notion d'incertitude, c'est-à-dire le moyen de reconnaître et prendre en compte le fait qu'un modèle puisse ne pas être adéquat à la situation concrète dans laquelle il doit être utilisé.

Nous avons alors proposé des approches alternatives, et par conséquent des façons inhabituelles de formuler les problèmes de la robotique autonome :

- *Sur le plan conceptuel,* nous avons proposé la notion de *représentation contingente* comme alternative à celle de représentation prédéfinie. La représentation contingente reconnaît explicitement le rôle fondamental du concepteur et de la notion d'interprétation, et considère que l'interprétation est toujours contingente au contexte.
- *Sur le plan méthodologique,* nous avons montré la nécessité d'isoler un cycle *incrémental* dans la construction d'un robot en complément des cycles de conception et d'adaptation. L'incrémentalité permet notamment de considérer l'imprévu comme un moteur de développement et non comme un phénomène parasite dans celui-ci. Nous avons proposé un embryon de méthodologie pour ce cycle, l'élément systématique étant l'adoption de certaines structures, les DEP, comme invariants de base du processus incrémental. Nous suggérons que les DEP constituent une forme de représentations internes.
- *Sur le plan théorique,* nous avons adopté la théorie probabiliste PaL (“Probability as Logic”), qui contient la logique traditionnelle mais formalise également le raisonnement incertain. Ce cadre théorique permet la gestion explicite de l'incertitude que nous voulions développer.

Comparé à cela, une certaine carence sur les plans technique et algorithmique est un peu inévitable au point où nous en sommes : les “solutions” que nous proposons aux difficultés de la robotique autonome consistant surtout à *reformuler* dans un premier temps les problèmes prioritaires, il est naturel que la maturation purement technique en soit un peu retardée.

B. Vers une réconciliation… ?

Notre approche peut être vue comme une tentative embryonnaire de réconciliation des approches fonctionnelles et structurelles concernant l'autonomie. Pour nous ces approches ne sont pas en compétition mais se situent sur des plans différents et complémentaires. L'approche structurelle nous semble seule apte à fonder une systématisation (formalisation) de la notion d'autonomie, en s'abstrayant de la nécessité d'objectiver l'analyse du concepteur humain par des positions philosophiques très fortes comme l'objectivisme. Toutefois, puisqu'en dépit de tout ce que nous pourrons faire ce sera en définitive un humain qui conduira la construction du robot, et son mode de pensée naturel est très nettement fonctionnel, la prise en compte de cette dimension reste indispensable.

La passerelle entre ces deux aspects de l'autonomie des robots est pour nous la notion d'interprétation. Des versions primitives de la représentation contingente sont apparues très tôt dans notre réflexion, comme autant de tentatives pour cerner et systématiser le rôle de l'interprétation dans la robotique autonome. C'est véritablement cette volonté de réconciliation entre approches fonctionnelles et structurelles qui a été la motivation essentielle et constante de notre travail. La problématique de l'imprévu et de l'incrémentalité n'est que le produit de cette volonté.

C. Perspectives

La notion d'imprévu comporte deux aspects, que nous avons jusqu'ici plus ou moins confondus mais que nous souhaitons mieux distinguer à l'avenir :

- Une forme d'imprévu réside dans le fait qu'une représentation interne puisse être réinterprétable dans différents contextes. Cet aspect de l'imprévu n'a pas de connotation de “surprise”, mais consiste à exploiter des représentations dans un cadre formel autre que celui ayant permis leur émergence. C'est cet aspect qu'illustrent les expériences de la bassine perturbée ou de la mire.
- Une forme d'imprévu réellement “créative”, en revanche, consiste à observer certains “effets de bord”, tels que l'évitement d'obstacle lors d'un comportement photophile ou la perturbation d'une trajectoire au passage d'un pas de porte, et ensuite en tirer parti pour faire évoluer le robot. Cette idée (l'imprévu et la contingence comme moteur de l'évolution) se rapproche de la conception moderne de l'évolution biologique, telle que décrite par exemple par S.J. Gould (1991, 1993).

Le premier aspect semble plus facilement systématisable que le second ; on devrait pouvoir décrire quelques formes génériques d'exploitation d'une représentation interne permettant une certaine variété de comportements (par exemple, reproduire l'aspect du comportement professeur ayant donné lieu à représentation, ou créer un prétraitement comparant les valeurs prédites et observées d'une même variable). Toutefois, ce premier aspect semble d'importance relativement limitée en comparaison du second (tirer parti de l'imprévu), qui fait l'essentiel de la richesse potentielle de notre approche.

Les directions que nous comptons prendre pour étudier plus avant ces deux aspects de l'imprévu sont alors :

- D'une part, entreprendre le développement incrémental de longue haleine du robot Khépéra, afin d'acquérir plus d'expérience et de dégager les priorités techniques pour mettre en oeuvre nos idées, et afin de juger de l'importance respective des deux aspects ci-dessus dans la pratique. Il s'agit en somme de “mettre la main à la pâte” pour identifier problèmes et solutions.
- D'autre part pousser plus loin la systématisation de l'acquisition et de l'exploitation de DEP. La voie qui nous semble la plus prometteuse pour cela serait de nature sélectionniste (Cliff 1992a, Steels 1993, Sims 1994). Cependant, la voie sélectionniste qui conviendrait à nos préoccupations ne semble pas pouvoir reposer sur un codage des caractéristiques d'un robot, mais sur un codage des caractéristiques de ses mécanismes de développement (analogue artificiel de l'ontogénèse).

Une autre perspective à plus court terme serait de perfectionner le petit système d'inférence probabiliste décrit au chapitre 4, de façon à pouvoir l'appliquer à des problèmes mettant en jeu de nombreuses variables, et en particulier au développement du Khépéra envisagé ci-dessus. Nous n'avons pas encore abordé de front ce problème, mais avons retenu quelques pistes à suivre, notamment les réseaux bayésiens (Pearl 1991, détails au chapitre 4) ou les techniques de Monte-Carlo (Neal 1993). Nous travaillons également en ce moment avec des physiciens et avec l'équipe de réseaux neuronaux du LIFIA pour identifier et creuser les liens

théoriques entre PaL, la physique statistique, les réseaux neuronaux et le principe de MaxEnt.[1] Nous espérons que cette étude permettra de réexprimer certaines techniques existantes (neuronales notamment) dans le cadre de PaL (les travaux de référence de cette démarche sont ceux de MacKay (1992)).[2]

Enfin, outre le développement de longue haleine de Khépéra, l'inspiration sélectionniste et le développement de PaL comme théorie fondatrice, nous aimerions intégrer la notion de clôture opérationnelle dans notre approche incrémentale. À terme, les comportements professeurs, actuellement supervisés (téléopérés ou programmés), devront eux-mêmes être générés par le comportement “propre” du robot, réalisant ainsi une clotûre entre perception et action. Nous pensons de plus (de façon très spéculative) que les problèmes de décision et de prétraitements, actuellements gérés par des procédés ad-hocs, seront largement dépendants de cette notion de clotûre, et nous souhaitons donc les aborder dans cette perspective. Étant donnée l'importance de ces problèmes, nous pensons que la voie de la clotûre opérationnelle, malgré sa difficulté, va s'imposer à une échéance assez courte.

---

1 De telles études ont déjà été menées, mais souvent dans le cadre de la physique ou de la statistique. La particularité de notre démarche est de se placer dans le cadre de l'IA : le caractère subjectiviste de PaL et son aspect “extension de la logique” sont les obstacles principaux du dialogue avec les physiciens ou les mathématiciens, mais c'est là justement la richesse de PaL que nous voulons garder tout en cherchant des inspirations techniques dans les applications plus traditionnelles des probabilités.

2 Actuellement, un certain nombre de techniques (connexionnisme, algorithmes génétiques), développées pour leur utilité empirique et formalisées en général comme faisant de l'optimisation de fonctions, restent en dehors du cadre unificateur de l'IA que constituent les sytèmes formels. Certaines approches “hybrides” (Giacometti 1992) cherchent à les faire coopérer — mais pas fusionner. Or, l'aspect numérique de PaL permet d'exprimer des problèmes d'optimisation comme “trouver la valeur la plus probable d'une DDP”. Cela semble un cadre intéressant pour établir des liens *théoriques* entre les démarches du raisonnement (qualitatif) et de l'optimisation (quantitative).

ANNEXES
et
BIBLIOGRAPHIE

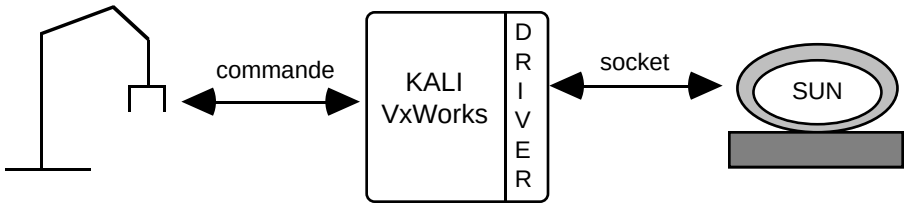
Annexe 1

Détails matériels sur les expériences

Commande du robot

Dans les expériences de la bassine lumineuse et de la mire, la cellule photoélectrique et la caméra sont fixées sur la pince d'un bras de robot SCEMI à 6 degrés de liberté (chapitre 4 figure 1, chapitre 5 figures 1 et 14). Le logiciel de commande est KALI 1.0, tournant sous l'environnement temps réel VxWorks.

Cette couche de commande étant usuellement utilisée par l'intermédiaire d'un logiciel de CAO (ACT) dont nous n'avions pas besoin, nous avons programmé un petit driver permettant de commander le robot depuis n'importe quelle station SUN (communications par sockets Unix), via quelques fonctions regroupées dans un module Lisp. L'architecture finale est celle de la figure ci-dessous.



Capteurs

La cellule photoélectrique (une résistance variable) est montée en pont diviseur de tension ; la tension mesurée aux bornes de la cellule est couvertie par un convertisseur analogique-digital intégré à l'armoire de commande.

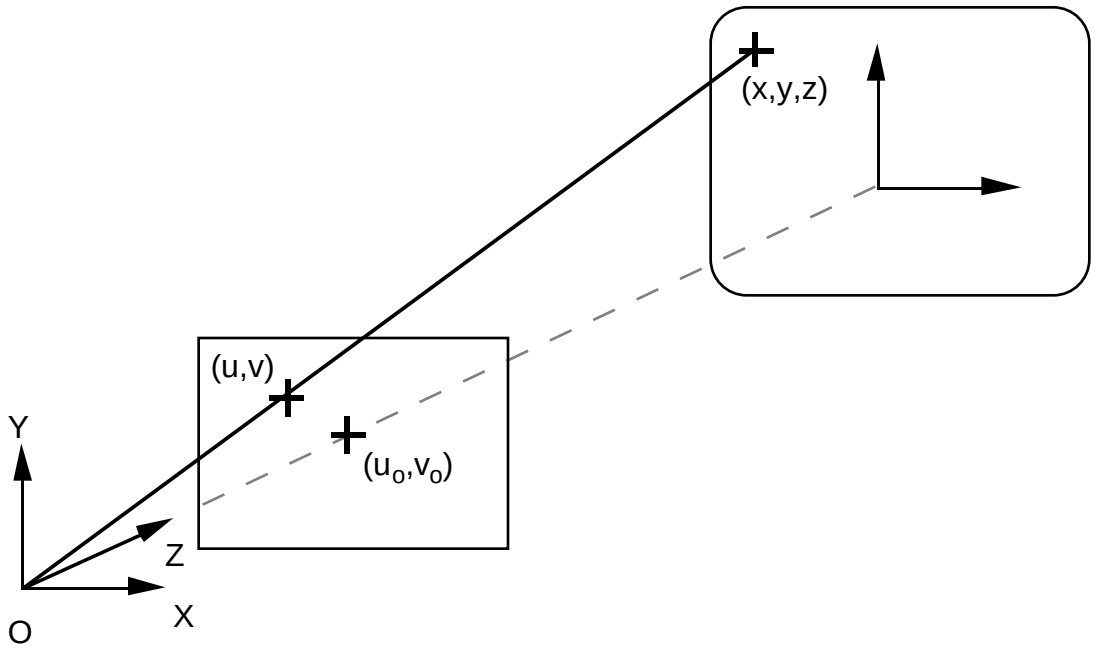
Le capteur proprioceptif (chapitre 4) est géré par la couche de commande (KALI).

La caméra est gérée indépendemment ; l'acquisition d'image se fait sur une carte FG-100 sur un SUN, un serveur permettant (via des sockets Unix) de contrôler cette acquisition et de transférer les images sur n'importe quel autre SUN. Le traitement de l'image se fait ensuite via des librairies développées par l'équipe vision du Lifia (MOVI).

Annexe 2

Le calcul analytique des prétraitements pour l'expérience de la mire

Nous montrons dans cette annexe le calcul du prétraitement “théorique” de l'expérience de la mire. La situation est celle de la figure suivante :



Un point (x,y,z) dans le repère O de la caméra se projette sur un point de l'image de coordonnées (u,v) par la formule :

$$u = \alpha_u (x/z) + u_o$$

$$v = \alpha_v (y/z) + v_o$$

avec α_u et α_u des facteurs d'échelle, et u_o et v_o les coordonnées dans l'image du point où passe l'axe optique.

En déplaçant la mire de Δz et de Δx, le même point de la mire se projette en :

$$u' = \alpha_u ((x+\Delta x)/(z+\Delta z)) + u_o$$

$$v' = \alpha_v (y/(z+\Delta z)) + v_o$$

On calcule que $(v-v_o)/(v'-v_o) = 1+\Delta z/z$. La variable `rapport` du chapitre 5 §B.2, que nous désirions pour un z fixé mettre en dépendance avec Δz, a alors été calculée comme la moyenne de la quantité $(v-v_o)/(v'-v_o)$ sur l'ensemble des points mis en correspondance entre deux images (voir figure 14 au chapitre 5).

Pour faire ce calcul, il est évident que nous avons utilisé des hypothèses très fortes (mire parfaitement perpendiculaire à l'axe optique et mouvement sans composante Δy). Ces hypothèses n'étant contrôlées que très approximativement (positionnement manuel de la caméra et de la mire), nous attendions clairement des incertitudes quant aux prédictions faites, d'où l'emploi d'un modèle probabiliste pour caractériser ces incertitudes.

L'ordonnée “centrale” v_o de l'image a été déterminée ainsi : sur une série de correspondances comme celle de la figure 14 (chapitre 5), nous avons repéré la limite v_{sup} des déplacements constamment positifs sur l'axe v et la limite v_{inf} de ceux constamment négatifs. v_o se situe entre ces deux bornes. Nous avons fait des essais et constaté une sensibilité négligeable de ce facteur sur les qualités de la DEP construite entre `trans` et `rapport`. Constatant cela nous avons alors pris pour v_o le milieu entre v_{inf} et v_{sup}.

Bibliographie

Agre, P. 1988 *The Dynamic Structure of Everyday Life,* PhD thesis, MIT.

Agre, P. & Chapman, D. 1987 Pengi: an Implementation of a Theory of Activity, *proc. of AAAI'87.*

Agre, P. & Chapman, D. 1989 What are Plans for?, AI-MEMO 1050a, MIT.

Barraquand, C. & Latombe, JC. 1990 A Monte-Carlo algorithm for path planning with many degrees of freedom, *IEEE robotics and automation.*

Blanes, C. 1991 *Guidage visuel d'un robot mobile autonome d'inspiration bionique, 2ème partie: implémentation opto-électronique et réalisation d'un prototype,* thèse INPG, Grenoble.

Bonabeau, E. 1992 Conceptions de l'émergence, workshop “émergence dans les modèles de la cognition”, ENST Paris.

Bouchon-Meunier, B. 1989 Incertitude, information, imprécision : une réflexion sur l'évolution de la théorie de l'information, *Revue internationale de systémique,* vol 3 n° 4, Dunod.

Boufama, B. 1994 *Reconstruction tridimensionnelle en vision par ordinateur: cas ces caméras non étalonnées,* thèse INPG, Grenoble.

Bouroche, JM. & Saporta, JM. 1980 *L'analyse des données,* Que sais-je?, PUF.

Bovet, P. & Benhamou , S. 1985 La clinocinèse: un mécanisme élémentaire de direction, in J. Paillard (ed), *La lecture sensorimotrice et cognitive de l'expérience spatiale,* CNRS.

Bovet, P. & Benhamou , S. 1988 Spatial analysis of animal's movements using a correlated random walk model, *Journal of theoretical biology,* n° 131.

Braitenberg, V. 1983 *Vehicles : experiments in synthetic psychology,* MIT Press/Bradford Books.

Brooks, R.A. 1986 Achieving artificial intelligence through building robots, AI MEMO 899, MIT

Brooks, R.A. 1989 A robot that walks: emergent behaviors from a carefully evolved network, *Neural Computations,* n° 1:2.

Brooks, R.A. 1990 Elephants don't play chess, in P. Maes (ed), *Designing autonomous agents: theory and practice from biology to engineeing and back,* MIT Press.

Brooks, R.A. 1991 Intelligence without Representation. *Artificial Intelligence,* vol. 47 (reçu en septembre 87).

Carpenter, G.A. & Grossberg, S. 1987 A massively parallel architecture for a self-organizing neural pattern recognition machine,*Computer vision, graphics, and image processing,* n°37.

Chaumette, F. 1990 *La relation vision-commande : théorie et application à des tâches robotiques,* thèse de l'Université de Rennes 1.

Cliff, D. 1990 The computational hoverfly: a study in computational neuroethology, in [SAB 90].

Cliff, D. & al. 1992a D. Cliff, P. Husbands & I. Harvey, Evolving Visually Guided Robots, CSRP 220, Univ. of Sussex, Brighton.

Cliff, D. & al. 1992b D. Cliff, P. Husbands & I. Harvey, Analysis of Evolved Sensori-Motor Controllers, CSRP 264, Univ. of Sussex, Brighton.

Connell , J.H. 1988 A behavior-based arm controller, AI MEMO 1025, MIT.

Cox, R.T. 1946 Probability, Frequency and Reasonable Expectation, *American Journal of Physics,* n° 17.

Dawid & al. 1973 A.P. Dawid, M. Stone and J.V. Zidek, Marginalization Paradoxes in Bayesian and Structural Inference, *Journal of the Royal Statistics Society* B35.

Deunebourg, JL. & Goss, S. 1990. Collective Patterns ans Decision Making, *Ecology, Ethology and Evolution,* Unit of Theoretical Behavioural Ecology, Universite de Bruxelles.

Dreyfus, H.L. 1979 *Intelligence Artificielle: Mythes et Limites,* Flammarion, 1984.

Dreyfus, H.L. 1993 La critique heideggerienne et l'approche husserlienne et searlienne de l'intentionnalité, *Intellectica,* n° 17, JM. Salanskis (ed.), Revue de l'ARC.

Dubois, D. & Prade, H. 1988 *Théorie des possibilités,* Masson.

Dubois, D. & Prade, H. 1994b Fuzzy sets — a convenient fiction for modeling vagueness and possibility, *IEEE transactions on fuzzy systems vol. 2 n° 1,* février 1994.

Dubois & al. 1994a D. Dubois, J. Lang & H. Prade, Possibilistic Logic, *Handbook of Logic in Artificial Intelligence and Logic Programming,* Gabbay & Hogger & Robinson (eds.), pp. 439-511, Clarendon Press.

Dubois & al. 1994b D. Dubois, H. Prade & P. Smets, Uncertainty versus partial truth : two distinct notions, rapport IRIT/94-40-R, Institut de recherche en informatique de Toulouse, novembre.

Dumouchel , P. & Dupuy , JP. 1983 (eds), *L'auto-organisation, de la physique au politique.* Colloque de Cérisy, Seuil.

ECAL 91 *Toward a practice of autonomous systems: proc. of the 1st european conference on artificial life (Paris 1991),* F. Varela & P. Bourgine (eds), MIT Press/Bradford books, 1992.

Edelman, G.M. 1989 *Neural Darwinism,* Oxford University Press.

Feller, P. & Touret, F. 19?? *L'outil: dialogue de l'homme avec la matière.* Editions Albert de Visscher.

Fikes, R. & Nilsson, N. 1971 STRIPS: a new approach to the application of theorem-proving to problem-solving. *Artificial Intelligence,* n° 2.

Fodor, J.A. & Pylyshyn, Z.W. 1988 Connectionism and cognitive architecture: a critical analysis, in S. Pinker and J. Mehler (eds), *Connections and symbols,* MIT Press.

Gardner, R.A. & Gardner, B.T. 1991 Feedforward: the Ethological Basis of Animal Learning, in [ECAL 91].

Giacometti, A. 1992 *Modèles hybrides de l'expertise,* thèse de l'ENST, Paris.

Gould, S.J. 1991 *La vie est belle,* Seuil.

Gould, S.J. 1993 *La foire aux dinosaures.* Seuil.

Harnad, S. 1990 The Symbol Grounding Problem. *Physica* D42.

Hogeweg, P. & Hesper, B. 1985 Socioinformatic Processes: MIRROR modeling Methodology. *Journal of Theoretical Biology,* n° 113.

Hornik & al. 1989 K. Hornik, M. Stinchcombe, H. White. Multilayer Feedforward Networks are Universal Approximators, *Neural Networks,* vol. 2.

Horswill , I.D. 1993 *Specialization of perceptual processes,* PhD thesis, MIT.

Jaynes, E.T. 1982 On the Rationale of Maximum Entropy Methods. *Proc. of IEEE'82.*

Jaynes, E.T. 1994 *Probability theory: the logic of science.* En préparation, version électronique disponible à bayes.wustl.edu (ftp).

Jaynes, E.T. 1978 Where do we stand on maximum entropy ?, in R.D. Levine & M. Tribus (eds),*The maximum entropy formalism,* MIT Press.

Kadane & al. 1986 J.B. Kadane, M.J. Schervish and T. Seidenfelt, Statistical Implications of Finitely Additive Probability, in Goell & Zellner (eds.) *Bayesian Inference and Decision Techniques,* Elsevier.

Kashioka, J.S. 1977 An Approach to the Integrated Robot with Multiple Sensory Feedback: Visual Recognition Techniques, *7th International Symposium on Industrial Robots,* Tokyo.

Khanna, T. 1990 *Fondations of Neural Networks,* Addison-Wesley.

Kohonen , T. 1988 *Self-organisation and associative memory,* 2nd ed., Springer-Verlag.

Latombe, JC. 1990 *Robot Motion Planning,* Stanford University.

Lozano-Perez & al. 1993 *HANDEY: a robot task planner.* T. Lozano-Pérez, J.L. Jones, E. Mazer, P.A. O'Donnell, MIT Press.

Lloyd, D. 1988 *Simple minds.* MIT Press/Bradford Books.

Maes, P. 1991 Learning Behavior Networks from Experience, in [ECAL 91].

Maes, P. & Brooks, R.A. 1990 Learning to Coordinate Behaviours, proc. of AAAI'90.

Marr, D. 1982 *Vision.* W.H. Freeman.

MacKay, D.J.C. 1992 *Bayesian methods for adaptive models,* PhD thesis, California Institute of Technology.

Malcolm, C. & Smithers, T. 1989 Symbol Grounding via a Hybrid Architecture in an Autonomous Assembly System, DAI-RP 420, Univ. of Edinburgh.

Mason, M.T. 1989 Robotics Manipulation: Mechanisms and Planning. in M. Brady (ed.) *Robotics Science,* MIT Press.

Mazer, E. & Miribel, JF. 1985 *Le langage LM,* Cepadues Editions, Toulouse.

McFarland, D. 1987 *Dictionnaire du comportement animal.* (editor), coll. Bouquins, Laffont, 1990.

Mel, B. 1990 *Connexionist robot rotion rlanning: a neurally-inspired approach to visually-guided reaching,* Academic Press.

Mekhnacha, K. 1995 Apprentissage et représentation sensori-motrice de l'évitement d'obstacle pour un robot mobile, rapport de DEA, LIFIA-IMAG, Grenoble.

Merleau-Ponty, M. 1942 *La structure du comportement.* 8ème éd., PUF, 1977.

Mondada & al. 1993 F. Mondada, E. Frenzi et P. Ienne, Mobile robot miniaturization: a tool for investigation in control algorithms, *Proc. of the 3rd International Symposium on Experimental Robotics,* Kyoto.

Neal, R.M. 1993 Probabilistic inference using markov chain Monte-Carlo methods, CRG-TR 93-1, University of Toronto.

Pearl, J. 1991 *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Reasoning,* 2nd ed., Morgan Kaufmann.

Pichon, JM. 1991 *Guidage visuel d'un robot mobile autonome d'inspiration bionique, 1ère partie : principes et simulation du robot,* thèse INPG, Grenoble.

Pierce, D. 1991 Tabula rasa learning of turn and travel actions, AI 91-157, Univ. of Texas, Austin.

Poincaré, H. 1902 *La science et l'hypothèse,* Champs, Flammarion, 1970.

Pomerlau, D.A. 1989 Alvinn: an autonomous land vehicle in a neural network, CMU-CS 89-107, Carnegie-Mellon University.

Reeke, G.N. & Edelman, G.M. 1988 Real Brains and Artificial Intelligence, in S.R. Graubard (ed.) *The AI debate: false starts, real fundations,* MIT Press.

Ribeiro & al. 1992 F. Ribeiro, JP. Barthès & E. Oliveira, Dynamic Selection of Action Sequences, in [SAB 92].

Richalet, J. 1991 *Pratique de l'identification.* ADERSA, coll. Traité des nouvelles technologies, Hermès.

Ring, M. 1992 Two Methods for Hierarchy Learning in Reinforcement Environments, in [SAB 92].

Robert, C. 1990 An Entropy Concentration Theorem: Applications in Artificial Intelligence and Descriptive Statistics. *Journal of Applied Probabilities,* september 90.

Rumelhart & al. 1986 *Parallel distributed processing, vol. 1.* D.E. Rumelhart, McLelland and the PDP research group (eds), MIT Press.

Ruspini, E.H. 1989 The semantics of vague knowledge, *Revue internationale de systémique,* vol 3 n° 4, Dunod.

SAB 90 *From animals to animats: proc. of the 1st international conference on the simulation of adaptive behaviour (Paris 1990),* J-A. Meyer & S.W. Wilson (eds), MIT Press/Bradford books, 1991.

SAB 92 *From animals to animats 2: proc. of the 2nd international conference on the simulation of adaptive behaviour (Honolulu 1992),* J-A. Meyer, H.L. Roitblat & S.W. Wilson (eds), MIT Press/Bradford books, 1993.

Shannon, C.E. 1949 *The Mathematical Theory od Communication,* University of Illinois Press.

Sims, K. 1994 Evolving 3D Morphology and Behavior by Competition, *Artificial Life IV proceedings,* R. Brooks & P. Maes (eds), MIT Press.

Smolensky, P. 1986 Information processing in dynamical systems: foundations of harmony theory, in [Rumelhart & al. 1986].

Steels, L. 1990 Towards a Theory of Emergent Functionality, in [SAB 90].

Steels, L. 1993 Building Agents out of Autonomous Behavior Systems, AI-MEMO 93-05, AI Laboratory, Université de Bruxelles.

Stewart, J. 1993a ed., *Intellectica,* n° 16, Revue de l'ARC.

Stewart, J. 1993b Au delà de l'inné et de l'acquis, in [Stewart 1993a].

Stewart, J. 1994a The Implications for Understanding High-level Cognition of a Grounding in Elementary Adaptive Systems, in P. Gaussier and JD. Nicoud (eds.) *From Perception to Action,* IEEE Computer Society Press.

Stewart, J. 1994b Un système cognitif sans neurones: les capacités d'adaptation, d'apprentissage et de mémoire du système immunitaire. *Intellectica,* n°18, A. Nguyen-Xuan et PY. Raccah (eds), revue de l'ARC.

Stone, M. 1970 Strong Inconsistency with Uniform Priors, *J. Am. Stat. Ass'n* n° 71.

Sutton, R.S. 1990 Reinforcement learning architectures in animats, in [SAB 90].

Takeyasu, S. 1977 An Approach to the Integrated Robot with Multiple Sensory Feedback: Construction and Control Functions, *7th International Symposium on Industrial Robots,* Tokyo.

Tyrrell, T. 1992 The Use of Hierarchies for Action Selection, in [SAB 92].

Tyrrell, T. & Mayhew, J.E.W. 1990 Computer Simulation of an Animal Environment, in [SAB 90].

Varela, F. 1988 *Connaître : les sciences cognitives.* Seuil

Varela, F. 1989 *Autonomie et connaissance.* Seuil

Varela & al. 1993 F. Varela, E. Thompson et E. Rosch, *L'inscription corporelle de l'esprit,* Seuil.

Webb, B. 1995 Using robots to model animals: a cricket test. To appear in *A State of the Art on Autonomous Robots and Artificial Live,* ???.

Winograd, T. & Flores, F. 1987 *Understanding computers and cognition.* MIT Press

RÉSUMÉ. L'objet de la robotique autonome est d'éliminer l'intervention du concepteur humain dans le fonctionnement d'un robot en environnement "complexe". Or, la programmation traditionnelle d'un robot repose sur l'utilisation de modèles dont le domaine de validité est assez restreint. Quand on sort du domaine de validité, on tombe sur le problème de *l'imprévu,* objet de cette thèse. Nous affirmons d'abord que l'autonomie d'un robot ne peut être obtenue sans une gestion systématique de l'imprévu, et que les approches habituelles de la robotique (hiérarchiques, comportementales, adaptatives) ne sont pas adéquates pour aborder cette question dans des environnements naturels non précisément contrôlés. Nous proposons alors trois pistes pour contourner ces limites. Sur le plan théorique, nous défendons la nécessité d'une reconnaissance explicite par un robot de sa propre ignorance, et donc d'une gestion systématique de l'incertitude, et adoptons pour cela une théorie de la logique probabiliste (Jaynes 1995). Sur le plan méthodologique, nous complétons le tandem conception-adaptation par une démarche incrémentale, i.e. la systématisation d'une évolution structurelle en réponse à certains imprévus. Nous mettons ainsi l'accent sur l'origine et la genèse des représentations plus que sur leurs performances. Sur le plan conceptuel, enfin, nous proposons la notion de "représentation contingente", qui définit une représentation non par sa fonction mais par sa structure : la capacité de représentation est intrinsèque à cette structure, mais son interprétation effective est contingente au contexte. Les problèmes liés à la représentation classique avaient conduit certains auteurs à rejeter la notion même de représentation, et avec elle un guide de conception incontournable. La représentation contingente est une tentative permettant d'aborder le problème de la conception au sein d'approches non encore exploitées en IA, telles que la clotûre opérationnelle.

Mots-clés : sciences cognitives, robotique, autonomie, représentation, incrémentalité, probabilités.

ABSTRACT. Robot autonomy will be achieved when robots can act in complex environments without the need of human intervention. However, the traditional methods of robot programming rely on models having very restricting conditions of validity. The problem of *inexpectation* arises when these conditions are not met in the real situation. We argue that robot autonomy can't be achieved without a systematic way for taking inexpectation into account, and we explain why the classical hierarchical, behavioural or adaptive approaches of robotics are too limited for tackling this problem in natural, not carefully controlled environments. We then suggest three paths for escaping some of those limits. Our first point is theoretical : a robot should be able to ackowledge and model its being partially ignorant of its world. For this we advocate the theory of “probability as logic” (Jaynes 1995) as a fundamental framework. Our second point is methodological : we propose an incremental approach for building robots, i.e. a systematic method for structural evolution, the motor of which is the occurence of unexpected events. The concern underlying this approach is the origin and genesis of representations more than their performances. Our third and last point is conceptual : we propose a notion of “contingent representation”, defining a representation by its structure rather than its function. The representational capacity is intrinsic to the structure, but the representational contents (interpretation) is context-dependant. The classical notion of representation had lead some authors to reject the very notion of representation — thus giving up an unescapable guide for design, too. Contingent representation is an attempt for tackling the problem of design within new approaches yet unexploited in AI, like that of operational closure.

Keywords : cognitive science, robotics, autonomy, representation, incrementality, probability.