

**TECHNIQUES PROBABILISTES D'INTEGRATION
ET DE CONTROLE DE LA PERCEPTION
EN VUE DE SON EXPLOITATION
PAR LE SYSTEME DE DECISION D'UN ROBOT**

THESE

soutenue publiquement le 8 juillet 1988
A L'ETABLISSEMENT TECHNIQUE CENTRAL DE L'ARMEMENT

pour l'obtention du grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité: Informatique

par

Xavier MERLO



devant la commission d'examen

Président

Georges Giralt

Rapporteurs

Guy Demoment
René Schott

Examinateurs

Ernst D. Dickmanns
Roger Mohr
Michel Pétré
Bertrand Zavidovique



D 136 024192 2

236024 1922

Institut National Polytechnique de Lorraine

CM 1988 MERLO, X.

**TECHNIQUES PROBABILISTES D'INTEGRATION
ET DE CONTROLE DE LA PERCEPTION
EN VUE DE SON EXPLOITATION
PAR LE SYSTEME DE DECISION D'UN ROBOT**

THESE

soutenu publiquement le 8 juillet 1988
A L'ETABLISSEMENT TECHNIQUE CENTRAL DE L'ARMEMENT

pour l'obtention du grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité: Informatique

par

Xavier MERLO



devant la commission d'examen

Président

Georges Giralt

Rapporteurs

Guy Demoment
René Schott

Examineurs

Ernst D. Dickmanns
Roger Mohr
Michel Pétré
Bertrand Zavidovique

Je voudrais remercier tout spécialement:

Roger Mohr, professeur à l'Institut National Polytechnique de Lorraine, pour avoir bien voulu diriger cette thèse en dépit de l'éloignement de mon lieu de travail, à l'Etablissement Technique Central de l'Armement d'Arcueil

Bertrand Zavidovique, professeur à l'université d'Orsay, conseiller scientifique du laboratoire Système de Perception de l'ETCA, pour les orientations et l'esprit qu'il m'a aidé à donner à cette recherche

Guy Demoment, chargé de Recherche CNRS et professeur à l'Ecole Supérieure d'Electricité et **René Schott**, professeur au Centre de Recherche en Informatique de Nancy à l'Institut National Polytechnique de Lorraine, pour avoir accepté la tâche ardue de lire en détail ce mémoire et d'être rapporteurs

Georges Giralt, directeur du Laboratoire d'Automatique et d'Analyse des Systèmes à Toulouse, **Ernst Dietrich Dickmanns**, professeur à la Hochschule der Bundeswehr à l'université de Munich, et l'Ingénieur Principal de l'Armement **Michel Pétré** de la Direction des Engins, pour me faire l'honneur de faire partie du jury.

Je tiens également à remercier **Alain Lanusse** pour ses avis éclairés, ainsi que tous les membres de l'équipe "multicapteur" pour leur collaboration. Parmi les stagiaires qui en ont fait partie, je suis plus particulièrement redevable à **Jérôme Dupont de Dinechin**, pour m'avoir aidé, ainsi qu'à **Olivier Dessoude** et **Yannick Vimont**, pour avoir été les premiers à utiliser les modèles que j'ai étudiés dans le cadre concret et "physique" du contrôle de la plate-forme multicapteur.

L'ambiance du laboratoire, à la fois de travail et de camaraderie, a certainement contribué à la motivation et la liberté de réflexion nécessaires.

Enfin et surtout, je ne peux manquer de citer le soutien qui m'a été apporté par ma famille et mes amis. Qu'ils trouvent ici le témoignage de ma plus sincère gratitude.

**TECHNIQUES PROBABILISTES D'INTEGRATION
ET DE CONTROLE DE LA PERCEPTION
EN VUE DE SON EXPLOITATION
PAR LE SYSTEME DE DECISION D'UN ROBOT**

Résumé

On distingue deux problèmes qui se posent pour la perception des robots de troisième génération: l'intégration des résultats de traitements et la reconfiguration des multiples capteurs et traitements, en fonction des tâches que nous considérerons plus particulièrement, la détection et la classification.

Un niveau décisionnel traitant l'aspect intelligent de ces problèmes est concevable, voire nécessaire, mais les perceptions peu sûres, la nécessité du temps réel imposent de réaliser des intégrations et des contrôles à bas niveau sémantique, incluant des techniques de traitement de l'incertain.

Il apparaissait donc important d'étudier les applications des techniques probabilistes d'estimation récurrente à des représentations type objet/attribut.

Après avoir posé le problème et exposé les fondements théoriques, nous ébauchons des solutions. Enfin nous analysons des exemples d'algorithmes et nous présentons des simulations dans des exemples d'école.

**PROBABILISTIC TECHNIQUES
OF THE PERCEPTION
INTEGRATION AND CONTROL,
TO BE IMPLEMENTED
IN A ROBOTIC DECISION SYSTEM**

Abstract

We distinguish two problems for the perception of third generation robots: integration of signal processing results, and reconfiguration of multiple sensors and of signal processing units, considering applications of special interest to us: object detection and classification.

A decision level dealing with the intelligent aspect of these problems is likeable, if not necessary, but uncertain perceptions and real time constraints lead us to perform integrations and controls at a lower semantic level, thanks to probabilistic techniques.

Therefore studying the applications of recursive estimation to object/attributes representations is worthy being considered.

We first try to provide a satisfactory definition of the problem. Then we outline some theoretical grounds and we formulate tentative solutions. Finally examples of algorithms are derived and analysed, and results of toy-world simulations are presented.

CHAPITRE I: INTRODUCTION

- 1.1 Description d'un système de perception
 - 1.1.1 La perception en robotique
 - 1.1.2 Classes de capteurs et de robots
 - 1.1.3 Avantages des systèmes multicapteurs
- 1.2 Problèmes posés par la perception des robots
 - 1.2.1 Problèmes temps réel
 - 1.2.2 Intégration multicapteur
 - 1.2.3 Contrôle perceptif
- 1.3 Environnement de la thèse et démarche adoptée
 - 1.3.1 Aspects pratiques et théoriques
 - 1.3.2 Approches décisionnelle et probabiliste
- 1.4 Problèmes posés par l'intégration et le contrôle de la perception
 - 1.4.1 Les modèles d'automatique classique
 - 1.4.2 Outils IA: objets, contextes, règles
 - 1.4.3 Problème du traitement de l'incertain
- 1.5 Plan de la thèse, contributions

CHAPITRE II: REPRESENTATIONS UTILISEES ET ARCHITECTURE DU SYSTEME MULTICAPTEUR

- 2.1 Une expérience de pensée: un robot arbitre de tennis
- 2.2 Représentations
 - 2.2.1 Nature des connaissances et des représentations utilisées
 - 2.2.2 Notion d'objet
 - 2.2.3 Modélisation de l'environnement
 - 2.2.4 Niveaux de représentation et incertain
- 2.3 Architecture du système de perception multicapteur
 - 2.3.1 Modélisation des capteurs et traitements
 - 2.3.2 Modèle hiérarchisé
 - 2.3.3 Boucles de contrôle hiérarchisées
 - 2.3.4 Techniques de contrôle et d'intégration

2.4 Exemples

2.4.1 Le robot arbitre de tennis: suite et fin

2.4.1.1 Problèmes soulevés

2.4.1.2 Intégration des différentes représentations

2.4.1.3 Contrôlée et focalisation de la perception

2.4.2 Exemple 2: une simulation multicapteur

2.4.2.1 Description

2.4.2.2 Problèmes soulevés et résultats

CHAPITRE III: MODELE PROBABILISTE D'EVOLUTION DE L'UNIVERS ET DES CAPTEURS

3.1 Les stratégies bayésiennes

3.1.1 Un jeu contre la nature

3.1.2 Application de la formule de Bayes à l'intégration d'observations

3.1.3 Exemples

3.1.3.1 Intégration d'observations d'attributs

3.1.3.2 Modèles gaussiens multimodaux

3.2 Filtrage et contrôle de la perception à partir d'un modèle stochastique markovien

3.2.1 Le filtrage ou estimation récursive

3.2.2 Le contrôle adaptatif, le "dual control"

3.2.3 Le théorème D.A.I. d'allocation des ressources

3.2.4 Retour sur les aspects pratiques du contrôle perceptif

3.3 Etude d'une application: les systèmes de poursuite multicible

3.3.1 Description du problème

3.3.2 Complexification de l'univers et adaptation aux cibles

CHAPITRE IV: EXEMPLES D'INTEGRATION ET DE CONTROLE DE LA PERCEPTION

4.1 Exemple d'utilisation de la notion de contexte: une extension du filtre de Kalman

4.1.1 Représentation du contexte sous forme de paramétrisation discrète

4.1.2 Le modèle

4.1.3 La densité de probabilité multimodale et les formules d'itération du filtre

4.1.4 Annexe: formules d'itération de Kalman classiques

4.1.5 Application et conclusion sur cet algorithme

4.2 Contrôle d'une fenêtre d'observation

4.2.1 Le modèle

4.2.2 L'estimation récursive

4.2.3 L'optimisation du contrôle

4.2.4 Les résultats

4.2.5 Commentaires, extensions, applications

4.2.6 Liaison avec un système d'intelligence artificielle

CHAPITRE V: SCENARIO D'INTEGRATION MULTICAPTEUR

5.1 Introduction

5.2 Modélisation de l'évolution de la scène et de la perception

5.2.1 Modèle d'évolution

5.2.2 Modèle de la perception

5.3 L'estimation récursive

5.3.1 Prédiction

5.3.2 Intégration

5.3.3 Contrôle

5.4 Résultats

5.5 Liens avec l'intelligence artificielle

5.5.1 Représentations utilisées

5.5.2 Approximations et développement de l'arbre des hypothèses

5.5.3 Liaison avec un système d'intelligence artificielle

CHAPITRE VI: CONCLUSION

6.1 Problème d'intégration et de contrôle de la perception

6.2 Liaison entre un système probabiliste et un système d'intelligence artificielle

CHAPITRE I: INTRODUCTION

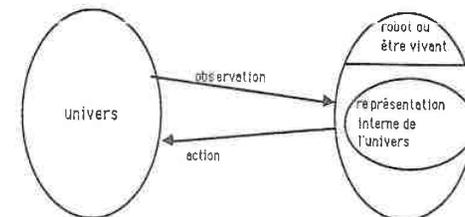
Les techniques probabilistes d'intégration et de contrôle de la perception sur lesquelles porte cette thèse s'inscrivent dans un cadre plus général de système perceptif d'un robot multicapteur. Dans ce chapitre d'introduction, nous commencerons par poser ce problème général et par isoler les sous-problèmes d'intégration et de contrôle perceptif. Puis, après avoir montré la nécessité d'utiliser des techniques d'intelligence artificielle, tout en conservant à un niveau plus bas une approche plus classique type automatique, nous décrirons les représentations de l'univers et des capteurs qui en découlent naturellement. Ces représentations feront l'interface entre le niveau décisionnel et le bas niveau probabiliste d'intégration et de contrôle de la perception.

1.1 Description d'un système de perception

1.1.1 La perception en robotique

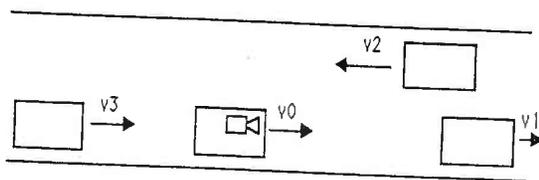
La perception est définie dans le dictionnaire (Le Robert) comme la fonction par laquelle l'esprit se représente les objets, ou l'acte par lequel s'exerce cette fonction (psychologie).

La perception est donc une fonction de création de représentation, qui se fait de manière active et non passive. On peut proposer le schéma suivant:



L'adéquation de la représentation de l'univers, et des modèles internes d'évolution et de perception se vérifie, et éventuellement se mesure par l'efficacité et les performances des actions qu'on en déduit. Tous les êtres vivants et les robots les plus évolués sont en bouclage permanent sur l'univers pour remettre à jour le modèle et les représentations associées, en fonction des variations observées. La perception est un problème en soi, qui est, sinon la principale limitation, du moins un des point-clés du développement de la robotique.

Prenons un exemple de système de perception dans le cas des systèmes évoluant dans des univers peu contrôlés qui nous intéresse: considérons un système robotisé pilotant une voiture sur une route. Pour cela, il dispose d'une caméra orientable et d'un microphone, qui sont les capteurs, et agit sur les commandes de la voiture (volant, accélération, freinage). Il s'agit alors d'un modèle classique étudié en automatique, où le système est constitué de la voiture et de la route. Le problème général du robot est de piloter la voiture, en évitant les sorties de route, les collisions.. et tout en optimisant la vitesse, la consommation suivant une certaine pondération. Le sous-problème de la perception est la construction à chaque instant du modèle de la scène afin de permettre au robot de remplir sa tâche, par un rebouclage de la perception sur les commandes.



modèle de la voiture robotisée

Pour cela, il dispose de divers traitements dont le but est d'analyser le signal recueilli par les capteurs, en vision, ou en acoustique, pour en extraire des informations significatives pour le problème, qui viendront s'intégrer en temps réel dans le modèle. Dans le système voiture+route, il s'agira par exemple de connaître:

- les paramètres intrinsèques de la voiture (position, vitesse ..); cela peut être fait par des capteurs internes dits **proprioceptifs** (alors que la caméra et le microphone sont des capteurs **extéroceptifs**)
- la position relative des bords de la route
- les autres véhicules ou objets sur la route avec leurs paramètres propres

On peut aussi vouloir reconnaître les autres véhicules, détecter les carrefours, les signaux routiers. Il sera également utile de disposer d'une carte à différents degrés de précision en

fonction du problème posé, ainsi que d'informations météorologiques pouvant influencer sur la conduite (pluie, verglas), et d'informations sur l'état de la route.

Le problème de la perception est donc celui de la construction de représentations internes de l'univers, connaissant:

- un modèle de cet univers, des objets qu'il comporte et de leur évolution
- un modèle des observations envoyées par les différents capteurs et traitements
- les différents buts et intérêts du système

Il s'agit alors d'un problème de traitement des observations, qui peut être considéré comme un problème inverse, en ce sens qu'on dispose d'un modèle naturel d'évolution et d'observation de la scène. On doit donc reconstruire les représentations à partir de ces observations, ou résultats des traitements appliqués sur la perception. De plus, dans la mesure où à chaque instant, doit être spécifié en temps réel une partie planification et contrôle des commandes capteurs et de la paramétrisation des traitements, en fonction de la prédiction d'évolution de l'état de l'univers, se superpose un problème d'automatique, au sens le plus large.

L'aspect actif de la perception est ainsi constitué de la construction et la manipulation de ces représentations, ainsi que de la commande des capteurs (tourner les yeux ou la caméra), du choix et de la paramétrisation des traitements (chez l'homme, focalisation de l'attention sur tel ou tel aspect de la scène).

Les représentations associées au modèle d'évolution de l'univers, parmi lesquelles on trouve les objets qui présentent un intérêt pour le système pour une tâche de détection ou de classification, sont décrites le plus naturellement par la notion informatique d'**objets** dotés d'**attributs**, par exemple des objets type véhicule, ou type route. Les objets font donc partie d'une classe, et sont caractérisés par la valeur de leurs attributs. Les classes peuvent être hiérarchisées en sous-classes (par exemple les véhicules peuvent être des camions ou des voitures..).

Il en résulte que les problèmes perceptifs qui se posent sont soit la mesure de tel ou tel attribut d'un objet, soit la détection de nouveaux objets et leur classification. Les objets sont également dotés d'un comportement, qui permet de prédire leur évolution (par exemple vitesse pour un mobile). Généralement l'intérêt du système porte sur un type particulier d'objet, ou sur la mesure de paramètres de cet objet (le robot vu plus haut s'intéresse surtout aux positions et vitesses relatives du véhicule qui est devant lui, afin d'éviter le choc!). Une fois donnés les moyens et les méthodes permettant de résoudre le problème posé, on progresse en définissant un critère de performance (rapidité..) sur les situations possibles.

Dans un univers robotique classique, le modèle de la perception et des capteurs est le suivant: l'univers du robot peut être décrit comme un ensemble de sources ponctuelles d'émission de signaux multispectraux (lumière visible, infrarouge, acoustique..) dans un univers géométrique 3D (ou plus si l'on introduit par exemple les vitesses). En général, les objets réels se résument à

plusieurs sources (ou même d'une infinité, pour la vision), qui s'organisent souvent en un niveau de représentation intermédiaire.

Les capteurs sont alors comme des fenêtres d'observation sur ce monde, définies par leurs limites géométriques et spectrales [El Jai], [Amat]. Les commandes de contrôle du capteur (par exemple pour une caméra: orientation, zoom, mise au point ..) déplacent cette fenêtre dans l'univers. Celle-ci constitue une ressource du système perceptif (dans l'accès aux observations), au même titre que les ressources d'un système informatique.

Les traitements dont dispose le robot lui serviront alors à extraire les différentes caractéristiques de la scène, ainsi que des objets qui s'y trouvent. On distingue les prétraitements, qui extraient des primitives de l'information sensorielle (contours régions... en vision, raies de fréquence, pic d'amplitude... en acoustique), des traitements d'un niveau supérieur, qui détectent les objets et leurs relations entre eux. Ces derniers reconstruisent un modèle de la scène lorsque la perception est complexe (vision), en se basant généralement sur les primitives, ou donnent directement des mesures de paramètres, à partir de l'image sensorielle d'une scène simple ou dont on a déjà construit un bon modèle (par exemple: mesure de la distance du véhicule précédent par stéréo, qui implique la localisation préalable de ce véhicule dans l'image).

1.1.2 classes de capteurs et de robots

Suivant le modèle précédent, on peut séparer grossièrement les capteurs et traitements associés en trois types:

- les capteurs de mesure simple, qui se contentent de donner la valeur d'un paramètre précis bien déterminé dans le modèle du système. On trouve dans cette classe beaucoup de capteurs proprioceptifs, d'asservissement de commande (à chaque fois que le modèle du système est bien connu)
- les capteurs renvoyant une image simple de la scène, dans laquelle en particulier seuls les objets significatifs apparaissent, ou apparaissent dans une scène relativement prévisible, et les bruits et les erreurs d'observation ayant un aspect statistique relativement simple, en particulier non corrélé dans le temps. Sont notamment dans cette classe, les capteurs de type radar ou sonar les plus classiques.
- les capteurs complexes, qui délivrent une véritable image de l'univers extérieur (par exemple visuelle, acoustique..), dans laquelle les objets perçus, bien que devant faire l'objet de traitements de perception, ne sont pas tous intéressants. Ils sont caractérisés par l'orientation et la taille de leur fenêtre d'observation et par leur bande spectrale. Ce sont typiquement des capteurs extéroceptifs, qui fournissent des signaux multisources motivant une véritable

analyse de scène tenant compte des phénomènes d'ambiguïté, masquages ou superpositions, l'exemple extrême, puisque l'on n'a plus affaire alors à une image d'émission, mais de réflectance, et où les sources ne sont plus aussi clairement discernables, les objets intermédiaires étant des contours, des régions, des points d'intérêts, des cartes de vitesse... Ces capteurs complexes sont donc plus difficiles à utiliser, mais permettent de bien observer l'univers externe dans toute sa variabilité. Ils peuvent en outre, dans la mesure où les informations fournies sont sûres, piloter d'autres capteurs ou traitements simples, qui mesurent des paramètres précis d'objets de la scène.

A ces trois classes de capteurs correspondent d'une manière non bijective trois générations robotiques. Une hypothétique quatrième génération capable d'analyser une scène quelconque reste du domaine du long terme, mais on prévoit néanmoins d'intégrer dans la troisième génération des débuts de capacité d'analyse de scène dans des univers relativement peu contrôlés.

- La première génération regroupe des automates sans capteurs reproduisant des séquences programmées. Il s'agit donc d'une répétition de tâches sans perception ni décision autres que d'éventuelles sécurités d'arrêt en tout ou rien.

- Les robots de deuxième génération possèdent une perception et une prise de décision minimale: le robot a la possibilité de se recalculer par rapport à de petites variations de la séquence programmée. La perception donne alors un écart par rapport à l'objectif, et l'on retombe sur un contrôle de type automatique dans des modèles très contraints en utilisant plutôt des capteurs simples du premier type. Il existe d'ores et déjà dans certains langages de robotique des extensions permettant de se recalculer grâce à des informations d'erreur simples, supposées exemptes de bruit. De plus, à une boucle de décision correspond en général une seule perception.

- Enfin, la troisième génération à venir est supposée capable de réagir en s'adaptant à des variations fortes quantitatives et qualitatives de l'environnement, dont elle possède un modèle élaboré, afin d'évoluer dans des univers peu ou non contrôlés, dans des milieux industriels ou militaires (robots autonomes). Elle devrait également pouvoir traiter des perceptions complexes, multisensorielles, multi-objets et incertaines.

La décision ne se traduit plus alors en de simples corrections d'erreur par rapport à une référence, mais doit être capable:

- de générer des plans d'action et de peser différents choix
- tout en pouvant réagir à chaque instant aux variations de l'univers ou du robot lui-même
- tout en restant directement contrainte par la spécification des tâches de haut niveau

1.1.3 Avantages des systèmes multicapteurs

La troisième génération de robots nécessite donc des capteurs complexes, seuls capables de rendre compte de la variabilité d'un univers peu ou non contraint, puisqu'ils restituent en une seule "image" de très nombreuses informations sur la scène, sans faire d'hypothèses a priori trop contraignantes sur celle-ci. Compte tenu de la complexité d'un tel système, des techniques de programmation évoluée (intelligence artificielle) seront nécessaires.

Cette dernière génération possède une perception multisensorielle potentiellement très souple et adaptable. Par ailleurs, chacune des images renvoyée par les capteurs complexes sera probablement analysée par plusieurs traitements. Nous allons étudier quelles pourraient être les caractéristiques de cette perception, à travers les cas des tâches de détection et de classification, qui trouvent une application dans le domaine militaire. Les avantages attendus sont:

- robustesse de la détection par utilisation de la redondance des informations (sur trois niveaux: possibilité de choix entre deux traitements, combinaison logique ou probabiliste des résultats de deux traitements, algorithme coopératif utilisant des informations multisensorielles)
- sensibilité et rapidité de la détection grâce à l'étendue du champ spectral
- possibilité de reconfiguration en cas de panne
- moindre sensibilité aux leurres dans un environnement hostile
- possibilité d'intégration des résultats de traitements frustrés, même renvoyant des informations peu précises et/ou incertaines
- faculté de traiter et de s'adapter à un grand nombre de situations, dans la mesure ou la partie contrôle (par opposition à perceptive), est également suffisamment souple.

Les inconvénients, i.e. les problèmes à résoudre sont l'accroissement important de la complexité du système et des scènes à traiter, et la nécessité de répondre rapidement à des changements brutaux et qualitatifs de situation, ce qui complique l'aspect temps réel du système.

Nous voyons donc que les problèmes de construction des représentations et de contrôle des capteurs et traitements vus plus haut deviennent encore plus critiques, à cause de la complexité des systèmes de perception de ces futurs robots, complexité répondant à la variabilité et au nombre des situations possibles. D'autre part, on voit dans l'exposé des avantages attendus une première indication de l'utilité de techniques de traitement de l'incertain (redondance, traitement des erreurs et des leurres, traitements frustrés).

1.2 Problèmes posés par la perception des robots

1.2.1 Problème temps réel

Une obligation pour le robot est donc de pouvoir réagir à chaque instant à une sollicitation imprévisible de l'environnement, compte tenu des possibilités de variation des univers peu contrôlés. Il doit pour cela maintenir en permanence des informations les plus complètes possibles sur la scène, en prévision d'une hypothétique décision urgente à prendre. Cela lui pose deux types de problèmes:

- l'un de puissance pure de traitement de l'information
- l'autre de gestion de cette information: l'utilisation de toutes les sources d'information et capteurs possibles, et la mise à jour en permanence des hypothèses sur l'état de l'univers, de manière à maintenir des informations disponibles sur la situation à chaque instant.

Le premier problème conduit à une architecture du système optimisant la rapidité du traitement: parallélisme physique par l'utilisation de plusieurs processeurs, spécialisés ou généraux, délocalisation de la décision dans des boucles rapides bas niveau de suivi de l'environnement, supervisées par une couche de plus haut niveau à une échelle de temps plus lente. Le deuxième problème temps réel conduit à utiliser plusieurs traitements et plusieurs sources pour acquérir la même information, les traitements moins précis et plus rapides suppléant provisoirement à l'absence ou au retard des résultats des autres [Bouvet]. On peut donc formaliser l'ensemble comme une configuration multi-experts, qui corrobore naturellement le parallélisme imposé par le problème de rapidité.

1.2.2 Intégration multicapteur

Les divers capteurs utilisés imposent naturellement encore plus de traitements travaillant les informations sensorielles: en effet, il ne s'agit pas de construire ex nihilo une machine traitant directement les données multisensorielles, mais plutôt un système intégrant les résultats de divers traitements existants. On rejoint ainsi dans cette conception multi-traitements multi-experts les modèles biologiques de perception (oeil humain [Wolfe], et les techniques les plus évoluées d'analyse de scène [Hanson & Risemann]). On suppose dès le départ que suffisamment de traitements existe pour fournir les éléments de décision nécessaire à la situation. L'objet de cette thèse n'est en effet pas de développer l'algorithmie dans ce domaine, hormis les éventuelles retombées en traitement de données des modèles et techniques développés. Si certains problèmes sont toujours non résolus, comme l'analyse de scène tridimensionnelle par la vision, pour de nombreux environnements peu ou non contrôlés, des traitements adéquats peuvent être trouvés. Le futur devrait permettre d'en élargir le spectre, de nombreuses voies s'avérant prometteuses (analyse d'images de distance, analyse du mouvement).

Pour rendre fonctionnel cet agrégat de traitements, sont nécessaires:

- un système d'intégration des résultats (quand il existe un espace des états où les intégrer), et notamment des incertitudes, afin de permettre des détections plus sûres et plus rapides (grossièrement, en faisant l'équivalent d'un ET multisensoriel sur les détections: la détection finale est plus sûre, ou d'un OU: elle est alors plus rapide, puisque couvrant un champ spectral plus vaste)
- une description symbolique efficace des traitements et des ressources perceptives (capteurs ou CPU partagés), permettant pour une configuration donnée un calcul rapide du coût que l'on cherche à minimiser (temps de calcul, incertitude des résultats). Dans les coûts étudiés, nous avons surtout considéré l'incertitude et l'imprécision des résultats, ainsi que l'adéquation avec la réalité de l'ensemble des hypothèses générées.

1.2.3 Problème du contrôle perceptif

Les systèmes automatiques ou robotiques du futur auront donc une perception nécessairement complexe, probablement multisensorielle et multitraitement (systèmes multicapteurs). Cette perception nécessitera de nombreux paramétrages, et l'utilisation au mieux de ce que nous appelons des ressources perceptives. Parmi celles-ci, on trouve essentiellement les fenêtres d'observation capteur (par exemple le champ de vision d'une caméra, le lobe d'un radar, mais aussi les fenêtres fréquentielles, électromagnétiques ou acoustiques), ainsi que les processeurs de traitement.

A chaque instant, un choix est donc à faire pour ces paramétrages et commandes: cela constitue le problème du contrôle de la perception [Zavidovique 85]. Ce problème peut être traité simultanément avec celui de la tâche à effectuer, c'est à dire en fonction de l'objectif du système global (perception + action, exemple au chapitre V), ou séparément, soit que la performance globale soit trop complexe à évaluer et qu'on lui substitue celles de chacun des deux sous-systèmes, soit qu'il s'agisse simplement d'un système de surveillance (exemple en 4.2), les parties décision et action étant alors du ressort d'un opérateur humain.

Dans le premier cas de l'étude globale, une approche de type automatique du problème aboutit à ce qu'on appelle le "dual control", c'est à dire que le contrôle influe à la fois sur les actions externes et sur la perception, l'optimisation étant réalisée sur une fonction de coût intégrant ces deux aspects. Dans le deuxième cas, il s'agit d'un problème de contrôle perceptif pur n'agissant que sur les capteurs et les traitements. Le critère d'erreur à minimiser dans le contrôle tient alors uniquement compte de l'adéquation de l'estimation réalisée à la suite des traitements avec la réalité, et de sa marge d'incertitude.

Ce contrôle perceptif présente quelques caractéristiques originales par rapport au contrôle pur:

- il présente un aspect plus immatériel, puisqu'il s'agit de maximiser une quantité d'information sur une scène donnée. De même, les "commandes" sont souvent internes: paramétrage et configuration des traitements. Néanmoins, dans certains cas simples, l'analogie avec des systèmes automatiques classiques est évidente. Dans un système de poursuite simple, on peut ainsi se contenter de reboucler l'écart de l'estimée de la cible sur la commande de l'orientation du capteur.

- de plus, on peut toujours augmenter le nombre de capteurs et faire des traitements en parallèle. On verra en effet au chapitre 3 que des observations supplémentaires ne peuvent qu'améliorer les performances du système. Bien entendu, dans un système réel, on sera toujours confronté à un problème de limitation d'une ressource perceptive.

- un aspect original est le traitement probabiliste de l'estimation, qui amène généralement à l'utilisation de méthodes d'estimation récursive. Hormis les traitements présentant peu d'erreurs, comme certains cas de poursuite cités ci-dessus, et les modèles d'évolution linéaires gaussiens (filtres de Kalman) à critère d'erreur quadratique, pour lesquels le principe de séparation nous permet de séparer le contrôle de l'estimation, on ne peut plus séparer les traitements probabilistes de la détection et de l'action. On a alors affaire à une densité de probabilité sur l'espace des états. L'optimisation de l'écart d'un estimateur (par exemple le maximum de vraisemblance) par rapport à la scène réelle (par exemple dans un scénario de surveillance) conduit alors à un calcul probabiliste sur l'espérance de cet écart, donc à des calculs probabilistes sur cette estimation probabiliste!

- d'autre part, dès que la scène à observer est relativement complexe (présence d'objets de plusieurs natures, possédant différents attributs), on a différents niveaux de qualité de la perception. Celle-ci s'affine alors au cours du temps à la fois numériquement (par convergence par exemple de l'estimation récursive sur la position et la vitesse) et qualitativement (reconnaissance du type de l'objet...). Au fur et à mesure de cette amélioration, on peut alors avoir un changement de l'objectif de la perception. La présence d'hypothèses discrètes (présence/absence ou type d'un objet, suivi sur plusieurs instants) conduit au bout de plusieurs étapes à faire des choix dans l'arbre des possibilités pour éviter l'explosion combinatoire de celui-ci, dès que l'on cherche à faire des prédictions sur un horizon temporel significatif (de nouveaux objets peuvent apparaître, ceux qui sont présents peuvent changer de comportement).

Parmi les problèmes apparentés classiques ou traités dans la littérature actuelle, outre les techniques de "dual control" et les filtres de Kalman déjà mentionnés, notamment dans leur version la plus évoluée de la poursuite multicibles [Blackman], on peut citer la cartographie de robots mobiles en scènes 2D ou 3D [Durrant-Whyte].

1.3 Environnement de la thèse et démarche adoptée

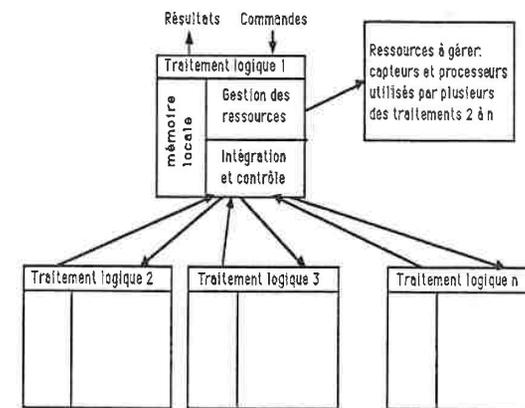
1.3.1 Parties pratiques et théoriques

Cette thèse se place dans le cadre d'un projet "machine multicapteur" du laboratoire "système de perception" de l'ETCA. Cette machine est conçue comme un moyen d'étude des problèmes de la perception exposés dans la partie précédente. Le projet a pour but la construction d'une plate-forme pourvue de divers capteurs tels que caméras visibles (3 caméras visibles sont prévues) et infra-rouge (1), une antenne de microphones, un télémètre, incluant des traitements spécifiques à chaque capteur, ainsi que des algorithmes spécialisés dans l'intégration de ces résultats, dans le contrôle coordonné des capteurs, et dans la planification de la commande des capteurs et des traitements.

Nous traiterons ici, à l'intérieur de ce thème plus général, des problèmes plus spécifiques d'intégration et de contrôle, sans aborder ceux que pose l'algorithmie des traitements, notamment pour les images. C'est à dire que nous supposons a priori que nous disposons de traitements fournissant des résultats suffisamment significatifs pour la tâche à effectuer. Nous nous concentrerons alors sur leur intégration et leur contrôle. Cette approche du contrôle perceptif pourrait par contre trouver une application en algorithmie, et donner ainsi une suite à ces travaux. D'ores et déjà, ce genre de considération existe en traitement d'images pour le contrôle de fenêtres d'applications des traitements [Cassou & Millour].

La réalisation de cette plate-forme pose un certain nombre de problèmes pratiques: nous avons vu ci-dessus qu'un modèle de traitement parallèle s'imposait naturellement, avec la possibilité de dérouler des boucles de suivi bas niveau simples et rapides, supervisées par une couche d'un niveau supérieur. Nous utiliserons donc un modèle classique hiérarchisé en plusieurs couches de représentation [Nii & al], où nous autoriserons des traitements parallèles et des prises de décisions locales. Cette architecture sera détaillée au chapitre II. Le problème n'est plus celui du séquençement des traitements, comme dans les modèles classiques d'analyse de scène, supposant généralement une structure monoprocesseur, mais celui des communications. Cette structure hiérarchisée simple permet de concentrer l'effort directement sur les problèmes d'intégration et de contrôle.

Par rapport à une décomposition simpliste en perception, décision et action, on a préféré un modèle hiérarchique de décomposition en sous-tâches, qui décentralise la décision dans des boucles de bas niveau [Kent & Albus]. Ce qui veut dire qu'au bas niveau, là où le flot des données est très important, et où il faut être très rapide dans les calculs, des capteurs ou des traitements simples peuvent être utilisés. Bien sûr, ces asservissements rapides sont pilotés par des traitements plus complexes, notamment au moment des transitions dans ces bouclages de bas niveau, qui ne sont pas pris en compte dans leur modèle trop simple du phénomène suivi. Par exemple, une boucle de poursuite vision ou autre peut être lancée et supervisée, ou paramétrée par un module de détection et d'analyse multiobjet de la scène.



architecture hiérarchisée choisie

On observe par ce biais un mélange entre la perception et l'action. C'est pourquoi il devrait être fructueux de s'inspirer à la fois des représentations d'analyse d'images ou de signaux [Hanson & Risemann], [Nii & al] et de programmation de robots [Haurat], [Latombe], [Demazaire], pour étudier l'utilisation de sources de connaissance multiples [Garvey & Fischler] et la réaction aux variations de l'environnement par réglage ou reconfiguration.

Pour valider les algorithmes développés, des études ont été conduites sur des simulations informatiques, en l'absence de la plate-forme multicapteur "physique". Cette démarche est parfaitement valide à condition d'observer une certaine rigueur dans la conception de ces simulations. On doit bien spécifier le modèle qui est simulé, ainsi que les critères qui permettent de vérifier l'adéquation et l'efficacité des méthodes utilisées. Cela est fait couramment, par exemple en intelligence artificielle avec des exemples d'école. Les simulations présentées ici auront de plus un caractère numérique, puisque la performance sera donnée par la valeur moyenne de la fonction de coût associée aux techniques d'automatique utilisées.

1.3.2 Relation entre l'approche décisionnelle et l'approche probabiliste

A côté de l'aspect pratique de ce projet multicapteur, sont étudiées de manière plus théorique des techniques logicielles destinées à traiter globalement le problème posé par le fonctionnement

coordonné des différents capteurs et traitements. Deux approches complémentaires sont envisagées, correspondant grossièrement à deux versions indépendantes d'un logiciel de gestion d'une machine multicapteur:

- des techniques de décision abstraite dans un univers non contrôlé, à base d'intelligence artificielle
- des techniques plus volontiers proche du signal, qui nous concernent ici, visant à réaliser la liaison entre les résultats des divers traitements, avec leurs incertitudes, leurs ambiguïtés, et les informations de haut niveau nécessaires à la décision.

Ainsi, si les méthodes d'automatique classique apparaissent insuffisantes, le temps réel et la pragmatique imposent cependant l'utilisation de boucles locales, mais supervisées par un niveau plus abstrait. Ces méthodes de contrôle (et plus précisément de contrôle des systèmes stochastiques) seront donc envisagées d'abord à bas niveau, en les adaptant à notre problème précis de perception et aux représentations IA choisies. A l'intérieur de l'équipe "multicapteur" du laboratoire, Alain Lanusse s'occupe plus spécifiquement de l'aspect décisionnel dans le cadre d'une thèse d'état [Lanusse], tandis que le présent mémoire se veut consacré aux techniques probabilistes et quantitatives de contrôle de la perception d'un robot, tout en utilisant des représentations de type intelligence artificielle. L'isolement de ce sujet précis peut apparaître à juste titre relativement arbitraire, et les exemples traités (en simulation sur ordinateur) auront un caractère académique et simplifié, sans être pour autant simplistes.

En fait, le caractère flou de la limite entre la partie décisionnelle et le traitement quantitatif de représentations parfois très élaborées, ainsi que la complétude des simulations réalisées font que seront également traités certains aspects de planification. Nous verrons que les techniques et les représentations IA (objets, arborescences), finissent par s'introduire naturellement dans l'aspect automatique, dès lors qu'il s'agit de contrôler la perception de systèmes (robot + environnement) complexes, et notamment multi-objets. On se retrouve en fait dans le cas plus général de la théorie des jeux et du calcul de fonctions d'évaluation. Le problème de traitement des événements ou des changements imprévus (non inclus dans le modèle) ne peut cependant être traité par ces techniques et reste du ressort de méthodes qui s'accrochent de connaissances vagues ou pragmatiques.

On est donc conduit à conjecturer que les relations entre les deux versions de prise de décision, celle utilisant des méthodes probabilistes d'automatique, et celle utilisant des techniques d'intelligence artificielle, ne se laissent pas facilement enfermer dans un schéma statique. Les problèmes posés pour la réalisation de l'intégration et du contrôle probabiliste de la perception, notamment dûs à la coexistence de représentations complexes avec des notions d'incertitude et à des problèmes concrets de commande capteur, ressortent comme suffisamment critiques pour être exposés en détail dans la partie 1.5 suivante.

1.4 Problème posé par l'intégration et le contrôle de la perception

1.4.1 Les modèles d'automatique classique

Pour résoudre le problème de la perception, nous avons besoin:

- d'un modèle de l'univers et de son évolution
- d'un modèle des capteurs, des traitements et de leur contrôle (les choix qui se présentent)
- d'un modèle de l'intérêt de l'observateur

Ces modèles sont donnés dans un contrôle du type automatique classique, et la plupart des systèmes rebouclés sur la perception existant aujourd'hui les utilisent: étant donné un modèle, souvent linéaire, du système, on cherche à asservir la commande sur le résultat en réinjectant l'erreur observée. Ce genre de contrôle, quoique très utile et même indispensable pour les problèmes de bouclage où le modèle est bien spécifié, a deux défauts principaux dans l'hypothèse d'une application aux robots futurs:

1) Les modèles usuels manquent de souplesse; un robot ne peut se satisfaire de tels modèles, son environnement étant en général complexe et incomplètement spécifié. La souplesse est au mieux introduite dans les seuls modèles de système stochastique par l'injection de bruit dans les équations d'évolution et d'observation du système, alors que des transitions brutales et/ou qualitatives peuvent se produire. La complexité et la variabilité de ces environnements ne peuvent donc être traitées complètement par des méthodes classiques, notamment dans des univers peu contrôlés ou même hostiles. On voit cependant apparaître de nouveaux modèles de contrôle, incluant des changements qualitatifs brutaux du système contrôlé sous forme de sauts poissonniens entre plusieurs types de comportement [Caines & Chen].

2) Ces techniques de contrôle mettent l'accent sur une optimisation de la commande, alors que souvent la question préalable se pose de savoir si la tâche que l'on cherche à accomplir est possible ou non, et comment la réaliser. Ensuite le critère à satisfaire pour un robot est la réussite de la tâche, tout en restant à l'écart de certaines zones de son espace des états, dangereuses pour lui-même ou pour son environnement. Le critère à maximiser peut alors tenir compte de l'écart à ces zones critiques, ou du pourcentage des cas de réussite.

Un système complet de contrôle de la perception devra donc utiliser des techniques d'intelligence artificielle, aussi bien pour ce qui est des représentations de l'univers (objets/attributs) que du contrôle proprement dit (planification, contrôle symbolique) [Zavidovique 85] et de l'intégration des résultats de plusieurs traitements (blackboard), d'où les deux approches choisies, l'une étant plus issue des techniques intelligentes de prise de décision, l'autre de techniques de traitement du signal plus quantitatives destinées à résoudre les problèmes de traitement de l'incertain et à piloter finement les commandes des capteurs. Nous pensons que ces deux approches sont destinées à converger, et c'est ce que nous allons illustrer dans ce mémoire en partant de la version probabiliste.

1.4.2 Techniques et représentations utilisées par l'IA

	Automatique	IA, robotique
représentations	paramètre d'états du système	objets / attributs scène, objets d'intérêt niveaux de représentation
modèle d'évolution de perception, et de la tâche	équations d'état, fonction de coût	modèle de comportement mixte: équations, modèle discret/continu, règles, prédicats, logiques
souplesse du modèle	rigide, fixé par les équations d'état souplesse limitée au bruit d'évolution et de perception	connaissances de différentes natures et de différents niveaux, parfois qualitatives et vagues (règles)

Comparaison des représentations, connaissances et techniques utilisées en automatique et en intelligence artificielle

Le laboratoire Systèmes de Perception de l'ETCA a une expérience dans le domaine du traitement d'image et dans les techniques de contrôle symbolique, étudiées notamment à l'occasion d'une expérience de robotisation d'une découpe laser [Foulloy & al]. Des techniques de contrôle de la perception ont été utilisées pour l'application des traitements en acoustique et en détection de mouvement [Cassou & Millour]. Bernard Burg étudie également l'apprentissage vu comme la génération de règles de comportement d'un système [Burg & Zavidovitch].

Le système de décision développé par Alain Lanusse fonde la reconfiguration d'un système de perception par détection de changement de contexte [Lanusse & al], [Lanusse]. Le contexte, notion souvent citée mais rarement utilisée en intelligence artificielle, indique ici un attribut ou une qualité globale de la scène ou de la situation qui se traduit par la prise en compte privilégiée de certains traitements, en fonction de ce qu'on s'attend à observer. Dans le modèle proposé par Alain Lanusse, le contexte, qui est donc une classification de l'état global de la scène, est détecté par un traitement séparé. A chaque contexte correspond des probabilités a priori d'occurrence des situations à discriminer. A partir de ces probabilités a priori, des résultats possibles de l'application des divers traitements sur la scène, et des coûts élémentaires des différents traitements dans le contexte courant, peut être compilé relativement rapidement un arbre de classification quasi-optimal des situations courantes, l'optimalité s'entendant au sens de la minimisation de l'espérance de la somme des coûts des traitements à appliquer pour discriminer complètement une situation. Chaque détection de changement de contexte entraîne

alors via ces tables de coût une refonte de l'arbre, afin de minimiser le coût moyen des classifications.

La définition d'un cadre d'intégration des résultats fournis par les capteurs et les traitements, doit donc permettre la mise à jour en temps réel le modèle de l'environnement utilisé par le niveau intelligent, qui ne connaît que des objets "significatifs" pour la tâche que le robot doit effectuer. En plus de la liste des objets ou des situations détectés avec les valeurs de leurs attributs lorsqu'elles ont été calculées, les différentes configurations des traitements permettant d'obtenir des précisions doivent être fournies à ce système de décision, avec leur coût, fonction inverse de l'efficacité escomptée.

Heureusement dans un grand nombre de cas les objets qui intéressent le système se traduisent aisément en terme d'objets physiques ou concrets (exemple: chars ou soldats ennemis). Mais d'autres notions plus globales ou plus complexes se rattachant à la notion de contexte, peuvent porter sur des situations (jour / nuit, anodin / menaçant) ou sur des comportements plus que sur des objets (par exemple: donner l'alarme en cas de comportements suspects près d'une zone sensible). Les principales tâches demandées au système de perception vont donc être des détections et des classifications, de manière à fournir les éléments nécessaires à la décision.

L'ensemble devrait ainsi fonctionner comme une base de données d'informations sur l'environnement, avec éventuellement une optimisation de l'accès ou l'indication de son impossibilité. L'élaboration de ce module de perception ne doit négliger ni les aspects dynamiques des primitives considérées (intégration temporelle et multisensorielle des données, couplage des capteurs, boucles de bas niveau), ni la possibilité pour le robot de réagir à des informations contextuelles très simples ne nécessitant que peu de traitement (par exemple changer le gain ou ouvrir le diaphragme de la caméra lorsqu'il fait trop sombre).

Un niveau intelligent est alors constitué de raisonnements logiques opérant sur des objets, des événements, à partir de lois qualitatives contenues implicitement dans le bas niveau, mais utilisées ici d'une manière plus efficace. Soulignons alors que cette approche IA classique peut également s'avérer trompeuse en ce sens qu'en passant de système traitant de logique pure à d'autres réalisant des bouclages sur le monde réel, on passe d'un univers virtuel que l'on peut manipuler et parcourir comme on veut, à un environnement où l'on est contraint par le temps réel. En particulier la notion de retour arrière ("backtrack"), qui correspondrait à une remontée dans le temps, n'est plus toujours possible dans la mesure où l'historique complet de tous les événements ne peut être conservé, et où, comme nous l'avons vu plus haut, la moindre erreur (petite au sens de la détection) peut avoir des conséquences catastrophiques, ou même définitives pour le robot si elle n'est pas correctement contrôlée. Il s'agit donc en fait de la substitution à une recherche en profondeur avec retour arrière, d'une recherche en largeur. Nous verrons que, de ce point de vue, les probabilités des différentes hypothèses permettent de classer celles-ci par ordre de priorité (chapitre III).

1.4.3 Problème du traitement de l'incertain

Les systèmes "intelligents" de planification robotique travaillent plutôt sur des informations relativement sûres; un des problèmes particuliers de la perception est au contraire la quasi-permanence d'observations incertaines. Cela est d'ailleurs apparu très tôt en intelligence artificielle, que ce soit dans les systèmes de diagnostic (MYCIN), ou dans ceux de traitement du signal (HEARSAY) qui incorporent des techniques d'évaluation du risque. On peut citer au titre de ce genre de techniques le traitement de l'évidence [Shafer], [Dubois & Prade] et le raisonnement flou [Zadeh], [Kaufman 73].

Dans ce contexte, l'intérêt du système multicapteur est d'étendre le spectre des données reçues, et de permettre de choisir, avec, bien sûr, toutes les nuances intermédiaires souhaitées, entre une détection rapide (on fait le OU multisensoriel de toutes les détections pour être sûr de ne rater aucun indice) ou sûre (on fait le ET pour combiner des observations indépendantes et augmenter la fiabilité de la détection), et ce à tous les stades. Nous avons choisi des techniques probabilistes classiques pour ce problème du traitement de l'incertain. On remarque à cet égard que celles-ci, longtemps dénigrées, ont connu assez récemment un regain de faveur dans les systèmes experts [Spiegelhalter]. L'emploi de modèles mixtes discrets et continus en automatique permet en outre d'accroître le domaine d'application de ces techniques, notamment pour les changements brutaux de conditions [Willisky & Jones], [Sworder & Chou].

Nous allons donc appliquer ces méthodes générales à notre problème particulier de contrôle perceptif, et en les adaptant à l'utilisation de représentations de haut niveau. Si nous en examinons les inconvénients, nous avons:

1) l'obligation d'avoir un bon modèle du système et de la perception, ou des statistiques suffisantes. En toute rigueur, doivent être spécifiés précisément:

- le modèle d'évolution probabiliste du système et des objets observés, avec la densité de probabilité de transition dans le cas du temps discrétisé
- le modèle de la perception: transformation mathématique aboutissant au signal plus bruit injecté, ou de manière équivalente, densité de probabilité conditionnelle d'observation. A mesure que l'évolution du système ne correspond plus au modèle, qu'il y a des événements ou des transitions imprévues, l'intégration des observations dans la densité de probabilité se trouve prise en défaut
- la fonction de coût, qui exprime l'intérêt du système et permet le contrôle des capteurs et des traitements par une optimisation simple dans le cadre du contrôle des systèmes stochastiques.

2) la lourdeur des calculs et la taille de l'espace des états, notamment lorsque le système est complexe, multi-objets.. (chapitre III).

Ce sont ces deux points qui ont conduit à l'éclosion des nouvelles techniques citées plus haut, et particulièrement de la théorie de l'évidence, qui demande moins de calculs et n'a pas besoin d'un modèle précisément spécifié.

Par contre, les avantages des techniques probabilistes, qui contribuent à leur regain de faveur, même dans le domaine des systèmes experts [Spiegelhalter], sont:

- 1) une base mathématique solide, qui permet le calcul de la densité de probabilité de la situation sur l'espace des états, ainsi que l'optimisation du contrôle
- 2) Les méthodes bayésiennes fournissent les stratégies optimales au sens de la théorie des jeux (chapitre III)
- 3) la possibilité d'utiliser des connaissances physiques et statistiques sur les objets, les capteurs et les traitements.

Ce dernier point a été décisif dans le choix de recourir à ces techniques probabilistes pour une intégration et un contrôle privilégiant le bas niveau, la perception étant un phénomène correctement modélisé sur le plan physique. Il s'agit en effet en général de transformations géométriques, avec injections de bruit additif sur les signaux de base. Le plus délicat est peut-être d'établir la densité de probabilité conditionnelle des traitements, notamment en ce qui concerne la vision, quoiqu'on trouve couramment par exemple des détecteurs de contours optimaux basés sur une modélisation mathématique des contours [Deriche], [Marroquin].

Toutes ces raisons, ainsi que les facettes théorique, pratique et opérationnelle du problème exposées précédemment justifient donc la formulation définitive du sujet:

"techniques probabilistes d'intégration et de contrôle de la perception en vue de son exploitation par le système de décision d'un robot"

1.5 Plan de la thèse et contributions

Les techniques probabilistes d'intégration et de contrôle de la perception sur lesquelles porte cette thèse s'inscrivent dans un cadre plus général de système perceptif d'un robot multicapteur. Dans ce chapitre d'introduction, nous avons d'abord posé ce problème général et isolé les sous-problèmes d'intégration et de contrôle perceptifs. Puis, après avoir montré la nécessité d'utiliser des techniques d'intelligence artificielle, tout en conservant à un niveau plus bas une approche plus classique type automatique, nous avons décrit les représentations de l'univers et des capteurs qui en découlent naturellement. Ces représentations feront l'interface entre un niveau de décision et le niveau probabiliste d'intégration et de contrôle de la perception, sujet de cette thèse.

Nous consacrerons le deuxième chapitre à l'étude des problèmes de représentation et de spécification de la tâche en robotique et plus spécialement pour les tâches de détection et de classification. Nous nous attacherons à décrire les représentations de l'environnement et des objets extérieurs utilisées généralement en robotique sous l'angle du système de décision du robot, qui correspond à ce qu'on appelle le niveau sémantique, par opposition à structurel ou à statistique. Le système de perception a alors pour tâche la construction de ces représentations à partir des résultats des différents traitements, ainsi qu'à partir d'informations a priori sur la scène. Une structure multitraitements s'impose donc naturellement. Se posent alors plus particulièrement les problèmes d'intégration et de contrôle de la perception, le dernier apportant une perspective originale au développement de robots de la nouvelle génération.

On verra au troisième chapitre que ces représentations (objets, contextes, heuristiques) issues des techniques d'intelligence artificielle et de robotique peuvent être étudiées sous un angle quantitatif probabiliste. Dans ce chapitre de bibliographie sur les modèles stochastiques, nous détaillons tout d'abord les techniques bayésiennes, ainsi que le principe d'une estimation récursive. Puis nous exposons le modèle de contrôle des systèmes stochastiques imparfaitement observés, avec en particulier la notion importante de "dual control", ainsi que les applications les plus élaborées dans le domaine de la poursuite multicible, dans lesquelles nous commençons à voir apparaître des techniques d'IA lors du développement des arbres de possibilités liés à l'évolution des systèmes discrets, et avec les heuristiques adaptées au domaine. Nous détaillons enfin les particularités du contrôle perceptif, avec en particulier un théorème sur l'allocation optimale d'une ressource dans un modèle stochastique simple grâce à des "indices d'allocation dynamique", connu sous le nom de "théorème des bandits à plusieurs bras".

Nous apportons une contribution à l'estimation des modèles mixtes discret-continu sous la forme d'un filtre de Kalman multimodal adapté (4.1), qui nous permet d'introduire une notion de contexte dans un modèle probabiliste, sous forme d'une paramétrisation discrète du modèle d'évolution et d'observation, qui varie ainsi par sauts poissonniens dans un ensemble fini de modèles linéaire-gaussiens.

Une simulation simplifiée d'un système de veille sur l'horizon (4.2) est traitée de manière satisfaisante par un système de contrôle optimal de surveillance, et illustre bien ce problème de contrôle perceptif présenté au chapitre précédent. Des résultats expérimentaux viennent conforter les approximations théoriques, et nous profitons à cette occasion de la simplicité du modèle simulé pour étudier expérimentalement le domaine de validité des stratégies bayésiennes, en faisant tourner la simulation avec des hypothèses fausses sur les probabilités a priori d'évolution et d'observation de la scène. Enfin nous évoquons les possibilités de liaison avec un système intelligent chargé du haut niveau.

Dans un dernier chapitre (chapitre V), nous présentons un autre modèle, dans lequel nous avons traité une scène plus complexe grâce aux mêmes méthodes: plusieurs capteurs, plusieurs

types d'objet, animés d'une vitesse dans un univers bidimensionnel. Puis nous abordons le problème de la prédiction et du contrôle sur des suites d'événements discrets, qui entraînent le développement d'arbres d'hypothèses, et donc là aussi l'utilisation de techniques symboliques d'intelligence artificielle.

Enfin nous conclurons sur les problèmes d'intégration et de contrôle de la perception que nous avons isolé, ainsi que sur les deux voies d'introduction de l'intelligence artificielle avec ces modèles probabilistes, en raison notamment des représentations d'objets et d'événements: dans une couche haut niveau, ou à l'intérieur même de l'estimation et du contrôle probabiliste.

CHAPITRE II

REPRESENTATIONS UTILISEES ET ARCHITECTURE DU SYSTEME MULTICAPTEUR

Dans ce chapitre, nous nous attacherons à décrire les représentations de l'environnement et des objets extérieurs utilisées généralement en robotique sous l'angle du système de décision du robot, qui correspond à ce qu'on appelle le niveau sémantique, par opposition à structurel ou à statistique. Le système de perception a alors pour tâche la construction de ces représentations à partir des résultats des différents traitements, ainsi qu'à partir d'informations a priori sur la scène. Une structure multitraitement s'impose donc naturellement. On verra au chapitre suivant que ces représentations issues des techniques d'intelligence artificielle et de robotique pourront être étudiées sous un angle quantitatif probabiliste dans une version simplifiée.

2.1 une "expérience de pensée": un robot arbitre de tennis

Nous allons illustrer ici les différents aspects architecturaux d'une machine multicapteur à travers une "expérience de pensée": la réalisation d'un robot arbitre de tennis. Cet exemple est intéressant car la sémantique du jeu de tennis (la signification du comportement des différents acteurs dans le jeu) est simple, puisqu'elle s'exprime spatialement à travers le comportement de la balle, et qu'elle fait à la fois appel à des données visuelles et acoustiques (cf les retransmissions de matches à la télévision!). Nous citerons également l'exemple d'une simulation que nous avons réalisé, à propos des interactions entre l'incertain et les structures de représentation.

Les intérêts du scénario qui va suivre (un échange dans une partie de tennis), sont donc les suivants :

- présenter une liaison simple entre les perceptions physiques (position des joueurs, de la balle) et la sémantique (analyse de la partie)
- permettre l'utilisation de perceptions redondantes pour interpréter le cours des événements (acoustique, vision multiple)
- permettre l'utilisation d'un modèle probabiliste simple (bayésien)
- présenter différentes sources de connaissance sur plusieurs niveaux (analyse physique de la parabole de la balle, connaissance de bornes sur les paramètres des trajectoires, évocation de quelques règles sur la tactique des joueurs)

De plus, le fait d'étudier une interprétation temporelle de la séquence d'événements permet de tester des aspects dynamiques spécifiques, moins présents dans, par exemple, l'analyse statique d'une image. On trouve ainsi dans le "temps réel" l'irréversibilité des prises de décision d'interprétation, même si on peut conserver plusieurs hypothèses en mémoire, et également l'utilisation d'algorithmes de prédiction pour affiner ses choix. Par rapport à un système robot réel, les hypothèses simplificatrices sont dans la schématisation de la réalité et des capteurs, par un scénario simple et des modèles probabilistes. Ce qui est intéressant dans ce modèle (qui n'a pas été programmé), c'est de voir comment pourrait être réalisée l'intégration multitraitements et multicapteur. Nous allons y retrouver en particulier tous les éléments exposés dans ce chapitre 2, et nous en tirerons des exemples afin d'illustrer notre propos.

Le but de ce robot logiciel est de suivre un échange d'une partie de tennis au sens des règles du jeu sans faire d'erreurs: arbitrage d'un échange et attribution du point, éventuellement de commenter quelques tirs et de donner des indications sur la force des joueurs. Ce scénario peut être interprété directement à partir d'une séquence de perceptions visuelles et acoustiques. La perception est modélisée sous forme de "traitements logiques" correspondant à des associations capteurs traitements, dont on se donne a priori les performances en fonction des paramètres de la simulation et de divers réglages: poursuite à partir d'images de vision, discrimination de chocs à partir d'informations acoustiques... Le programme se décompose en deux modules: le premier simule la partie et le second en interprète la perception.

La partie se déroule naturellement sur un terrain standard, avec lignes, filet et les différentes zones. Les joueurs A et B sont représentés par leurs coordonnées dans le plan, tandis que la balle possède six attributs scalaires: position et vitesse tridimensionnelles. La partie peut être simulée par programme, par modélisation au cours du temps de la variation des positions des joueurs et des position et vitesse tridimensionnelles de la balle. Par exemple, à la réception d'une balle, un tirage détermine si le joueur manque la balle ou non, en fonction de la distance initiale à la trajectoire et de la durée de celle-ci. Le tir se fait ensuite en suivant des considérations tactiques élémentaires: on "smash" si la balle est haute, on lobe si l'adversaire est près du filet, on tire dans les trous. Une imprécision aléatoire sur le tir, éventuellement fonction d'un attribut "force du joueur", provoque des balles out ou dans le filet. Un attribut "agressivité" pourrait

aussi modifier le tir d'un joueur (tir tendu au ras du filet, tir dans les coins plus risqués). Le comportement des joueurs a donc une certaine complexité, en fonction des paramètres de la situation (position de la balle et du joueur adverse) et de ses propres attributs.

Dans la partie analyse de scène, symétrique de la simulation, mais bien sûr indépendante, on étudie seulement un échange (échange de balles jusqu'au point pour l'un des deux joueurs, sans la partie service). Celui-ci peut alors être décomposé en intervalles de temps correspondant à un trajet libre de la balle, assimilé à un fragment de parabole. Ces intervalles sont séparés par des événements définis par les règles du jeu lors des divers chocs de la balle: sur une raquette, sur le filet, sur le terrain, out. Chaque événement correspond de ce fait à une perception visuelle et acoustique précise, mais doit aussi être interprété en fonction du passé: un rebond n'a pas la même signification suivant qu'il suit une frappe ou un autre rebond. La succession de ces événements, modélisés par un automate à états finis décrivant les règles du jeu, constitue une représentation symbolique de l'échange de balle suivi. Un niveau supérieur correspond à une description qualitative des joueurs (force, tactique ...). Nous allons donc tout au long de ce chapitre détailler ces représentations et les traitements associés, en y puisant les exemples illustratifs de notre propos.

2.2 Représentations

2.2.1 Nature des connaissances et des représentations utilisées

On peut grâce à une syntaxe du comportement de la balle et des joueurs dans cet exemple de robot arbitre de tennis, compter les points, et déterminer la signification d'une partie. D'autre part, l'arbitrage est une tâche riche en événements imprévus qui peuvent interrompre la partie, ou la modifier et qui nécessitent des techniques inspirées de l'intelligence artificielle, si on veut traiter complètement le problème. Nous allons ici développer ces deux aspects, afin d'effectuer une première distinction entre des modèles précis travaillant dans un espace des états du système, et les modèles utilisant des connaissances plus vagues et empiriques.

Les techniques classiques d'automatique travaillent sur un espace des états du problème, défini au départ par le concepteur du système grâce à des connaissances physiques du phénomène étudié. En robotique de troisième génération, l'environnement rencontré comportera également des structures de représentation que nous qualifierons de connaissances précises, parce que relativement figées, et pouvant être ramenées à une position dans un espace des états: une situation donnée peut être spécifiée de manière incertaine, par exemple par des probabilités, mais l'ensemble des situations est spécifié de manière précise et exhaustive dans cet espace.

Bien entendu, nous essaierons d'utiliser le plus possible ce genre de représentation, ce qui sera relativement aisé lorsqu'il s'agira, comme dans les systèmes de perception, de connaissances physiques sur par exemple la position des objets, ou la fréquence d'émission de sources de

rayonnement, donc sous-tendues par un modèle précis. Malheureusement ce genre de connaissance ne suffira probablement pas dans un univers non contrôlé ou partiellement contrôlé. En effet, dans un tel système, d'une part on ne pourra pas déterminer toujours les probabilités a priori d'occurrence de certains événements. D'autre part, il y aura toujours un phénomène imprévu qui ne s'intégrera pas précisément dans le modèle. Dans l'exemple de la partie de tennis, il peut s'agir du comportement aberrant d'un joueur, de la disparition hors du terrain de la balle, de la pluie, de la tombée de la nuit, voire de l'irruption d'intrus sur le court...

On s'aperçoit donc que:

1) pour une tâche donnée (l'arbitrage), le robot n'a pas besoin de comprendre parfaitement la scène, tant qu'il ne commet pas d'erreurs. Il a juste besoin de pouvoir déterminer des catégories de situations relatives à sa tâche, ce qui se passe d'une manière plus précise dans certaines ne le concernant pas. Le problème du système de décision est de pouvoir obtenir des indices précis permettant de détecter le passage d'une catégorie de situation à une autre.

2) par contre, il faut absolument éviter que le système "tourne à vide", c'est à dire qu'il fonctionne sur un modèle incorrect de la réalité: par exemple, si le robot compte des points au passage d'un pigeon (!). Pour éviter cela, il paraît nécessaire de disposer, au dessus d'un ensemble de modèles bien précis correspondant chacun à un contexte donné, c'est à dire à une catégorie de situations nécessitant le même comportement, une couche logicielle de décision. L'objectif de ce système est de piloter le changement de modèle, ou comme on le verra, dans un modèle probabiliste, de pondérer les différents modèles, tout en accumulant le maximum d'informations vagues. Il peut par exemple s'agir d'informations de bon sens destinées à détecter la cohérence de la situation (bornes sur les vitesses des joueurs et de la balle, sur les délais de réactions..).

Ce qui différencie ces connaissances des représentations des modèles sous-jacents, c'est qu'elles ne peuvent s'intégrer dans un cadre précis. Il n'est en effet pas possible de décrire exhaustivement tout ce qui peut se passer. On ne peut qu'accumuler des connaissances pragmatiques sur les événements possibles, et définir des niveaux de cohérence, pour lesquels on est sûr de rester dans un modèle donné, sans être forcément capable de décrire ce qui se passe exactement. On peut comparer ce problème à celui de la sécurité d'une centrale nucléaire: on dispose d'une part de modèles partiels de chacun des éléments, et d'autre part on se donne des catégories ou contextes globaux de fonctionnement. Pour optimiser la sécurité de l'ensemble, on doit optimiser celle de chacun des éléments, prévoir le maximum d'événements possibles (chute d'avion, tremblement de terre..) dans les calculs, et définir des niveaux de sécurité constituant une garantie contre l'imprévu.

Parmi les événements pris en compte ("précisés"), il reste un pourcentage non nul de catastrophes (par exemple un accident nucléaire majeur tous les 5000 ans). Il reste également l'imprévu simple (tout ce qu'on n'a pas prévu), ainsi que tous les problèmes qu'engendrent les

comportements intelligents, qui peuvent être imprévisibles ("erreurs humaines"), ou même malveillants. Par la suite, nous nous limiterons à des modèles précis de la réalité, en nous référant à la thèse en cours d'Alain Lanusse [Lanusse][Lanusse&al], et à la littérature spécialisée dans ce domaine de l'informatique symbolique, pour ce qui est des techniques de décision.

Nous utiliserons cependant des représentations complexes d'intelligence artificielle comme celle d'objet, voire même la notion de contexte. Dans une certaine mesure, une connaissance imprécise peut être incluse dans un modèle précis sous forme d'une probabilité a priori "estimée". Notre choix d'utiliser des techniques probabilistes comme unification entre les résultats des traitements et un système de décision intelligent est donc cohérent.

2.2.2 Notion d'objet

La notion d'objet a été introduite par Minsky, avec les "frames" [Minsky], dans laquelle un objet d'un type donné a une certaine structure, qui se traduit par une liste d'attributs. Ceux-ci permettent de prendre en compte toutes les caractéristiques d'un objet. Les objets de l'environnement sont ainsi naturellement représentés, de manière à pouvoir répondre rapidement à toutes les requêtes perceptives du système de décision. La scène elle-même est une "frame," ce qui permet de définir des attributs globaux, qui peuvent par exemple intégrer une notion de contexte. On doit également trouver privilégiés dans les attributs les aspects tridimensionnels: position, vitesse, forme, et plus généralement les grandeurs et les indices pertinents, ainsi que la notion de sous-objet composant ou partie du précédent.

Ces attributs sont des caractéristiques de l'objet qu'ils précisent. Leur valeur peut être simplement

- booléenne: dans ce cas elle indique simplement la présence ou l'absence de la caractéristique indiquée par l'attribut
- entière, réelle: l'attribut est précisé par un nombre, par exemple: vitesse, position...
- symbolique: précise une caractéristique parmi un ensemble de valeur (couleur, comportement), ou même sous forme d'arbre

Dans notre exemple de la partie de tennis, on utilise très naturellement des représentations objet/attribut pour les joueurs et la balle:

balle:

- position et vitesse 3D et incertitude

joueurs A et B:

- position et vitesse 2D, et position de la raquette
- éventuellement attribut d'agressivité ou de force caractérisant son jeu

Cette représentation objet/attribution permet de traduire n'importe quelle description structurelle d'un objet (arbre et/ou .), puisqu'on peut fabriquer cette structure avec les attributs ad hoc. On trouve ainsi dans les représentations utilisées dans le robot arbitre:

échange:

-suite de couples (trajectoire , choc)

trajectoire:

-paramètres temporels et spatiaux du segment de parabole, plus leur incertitude

choc:

-instant de la détection
-classification (raquette, sol, filet)
-position et vitesse de la balle
-incertitude de la détection

Dans les langages de programmation dits "objets" permettant d'utiliser simplement ce type de description, la notion d'héritage permet en outre de définir simplement des types d'objet à partir d'un type plus général, par exemple le type "paquebot" à partir du type "bateau". De plus à ces objets sont associés des actions qui sont activées par l'envoi de messages sur un objet d'un type donné.

Pour donner un exemple, définissons le type d'objet "objet-2D", qui sera décrit par son nom, sa position et sa vitesse (bidimensionnelles):

type:

objet-2D

attributs:

position-x
position-y
vitesse-x
vitesse-y

A cet objet nous pouvons associer une action "avance". L'application de cet action ou "message" à un objet de type "objet-2D" aura les conséquences suivantes:

position-x ← position-x + vitesse-x
position-y ← position-y + vitesse-y

Nous pouvons également créer les sous-types "joueur" et "balle" de "objet-2D":

type:

joueur

sous-type de:

objet-2D

autres attributs:

position-raquette
agressivité

type:

balle

sous-type de:

objet-2D

autres attributs:

position-y
vitesse-y

Naturellement, le sous-type "balle" impose une modification évidente de l'action du message "avance". Par défaut, ce message déclencherait en effet l'action associée au type hérité "objet-2D". Nous pouvons alors, après la modification nécessaire, créer les deux objets joueur-A et balle-jaune, de types respectifs "joueur" et "balle", et envoyer des messages à ces objets, messages qui sont en fait des fonctions spécifiques à chaque type ("fonctions génériques") pour obtenir les effets désirés:

(avance joueur-A)

(avance balle-jaune)

dans la syntaxe de l'extension objet du langage Lisp (flavors, [Moon]), avec laquelle nous avons travaillé. Le même message est pris en compte d'une manière spécifique par chaque type d'objet. Ici, on se contente de calculer la nouvelle position après un intervalle de temps unité par un simple incrément de la vitesse sur la position, d'une manière bidimensionnelle pour les joueurs, et tridimensionnelle pour les balles. Mais ce message pourrait aussi bien être envoyé à une particule microscopique Brown-1, dont la nouvelle position serait alors choisie de manière aléatoire: tout dépend de la définition de l'action associée au type d'objet, l'action par défaut du type hérité étant effectuée en l'absence d'une autre définition. Nous voyons donc que ce type de codage va nous permettre de décrire la scène en termes compréhensibles par le système de décision du robot, qui pourra raisonner sur et en fonction d'objets connus par lui. Un exemple d'instruction qui pourrait alors être donnée à un robot ayant perdu la balle de vue est:

si objet détecté = balle OK

si objet détecté = oiseau ou avion continuer à chercher la balle

La classification de l'objet détecté, en balle ou autre, peut venir d'une reconnaissance des formes directe, ou de la précision d'attributs (vitesse...) permettant de discriminer les objets.

```

Objets:
  More above
  #<OBJET 25023504> #<OBJET 25016643> #<OBJET 25005730> #<OBJET 17310316> #<OBJET 17310246> #<OBJET 17310264> #<OBJET 17310344>
  #<OBJET 25023504>
  #<OBJET 25016643>
  #<OBJET 25005730>
  #<OBJET 17310316>
  More below
  #<OBJET 25005730>
  An instance of OBJET. #<Message handler for OBJET>
  X-POSITION: 407
  Y-POSITION: 479
  ANGLE: 0
  More below
  #<OBJET 17310316>
  An instance of OBJET. #<Message handler for OBJET>
  X-POSITION: 419
  Y-POSITION: 47
  ANGLE: 0
  More below
  #<OBJET 17310246>
  An instance of OBJET. #<Message handler for OBJET>
  X-POSITION: 407
  Y-POSITION: -43
  ANGLE: 200
  X-VITESSE: -8
  Y-VITESSE: -16
  R-VITESSE: 8
  FORME: ((7 0) (7 240))
  COULEUR: unbound
  MDX: unbound
  COMPORTEMENT: "tout-droit"
  Bottom of object
06/24/87 16:20:33 merlo USER: fvj

```

exemples d'objets créés sous forme de flavors dans le langage lisp

Enfin, cette notion informatique d'objet ne correspond pas forcément à un unique objet physique, et peut être composée d'un ensemble ou d'une structure de sous-objets, liés entre eux par des relations, et dont on peut définir le comportement d'une manière globale (par exemple formation de véhicules dans le domaine militaire).

2.2.3 Modélisation de l'environnement

Les problèmes auxquels nous nous sommes attachés dans cette thèse concernent surtout une modélisation de robot de surveillance, et sont donc la détection et la classification. Dans le domaine militaire, par exemple, ces tâches tirent déjà parti d'informations multisensorielles ou multitraitements pour analyser des scènes complexes multiobjets [Moravec & Nasburg]. La détection consiste à extraire le signal émis par l'objet de la scène ou du bruit de fond, et ce le plus rapidement possible. La classification précise la nature de cet objet, (par exemple un véhicule ou un mobile est-il une voiture ou un camion?), et en détermine certaines caractéristiques sémantiquement importantes (présente-t-il les caractéristiques d'un ami ou d'un ennemi?). De plus, il peut parfois être nécessaire de préciser la position de la cible par une localisation, à cause des insuffisances propres à chaque capteur (absence d'indication directe de

distance en vision, imprécision en acoustique). On peut donc constater que la représentation objet/attribut est la manière la plus naturelle de poser ces problèmes, puisque la traduction de ces derniers en terme d'objets est immédiate. La figure vue plus haut montre des objets qui ont été utilisés dans une simulation développée sur une machine spécialisée en intelligence artificielle, appelée machine lisp. Cette machine intègre d'origine une représentation objet appelée "flavor" dans le langage de programmation lisp.

Dans une représentation haut niveau de la scène, nous nous proposons donc d'intégrer en particulier les données suivantes, grâce à cette description objet/attribut. La scène peut être définie comme étant une description symbolique physique de l'environnement inclus, ou ayant été inclus dans le champ des capteurs. Elle va se subdiviser en trois composantes:

- l'état de la machine, qui donne le réglage actuel des actionneurs et éventuellement la configuration des traitements. On peut en déduire, éventuellement de manière dynamique, les coûts des différentes actions possibles. Pour le système d'intégration, l'état de la machine et des capteurs s'agence la description générale de l'environnement.

- le contexte, qui est une suite d'attributs, globaux (jour/nuit) ou localisés par zones (classification du clutter en radar) de la scène. Des exemples type en sont les caractéristiques générales de l'environnement des sources et des conditions de propagation de l'information perçue par chaque capteur (niveau de luminosité, bandes perturbées par le vent en acoustique), ou une analyse qualitative globale de la scène (dans un scénario militaire, situation de veille ou de menace..).

- une liste d'objets, supposés tridimensionnels, rigides, et rayonnant dans différents spectres sensoriels (visible, acoustique..). Chaque objet possède une liste d'attributs, parmi lesquels des indications du genre position, vitesse, son, forme que l'on peut remplir par des indications plus ou moins précises (valeur exacte ou intervalle de variation) et même éventuellement des données de bas niveau (accès possible vers une mémoire d'image destiné par exemple à un traitement complexe de reconnaissance). Peuvent également être spécifiées des relations avec d'autres objets, sous forme de pointeurs étiquetés (masque, partie de ...), ainsi que des groupes, cortèges ou formations d'objets.

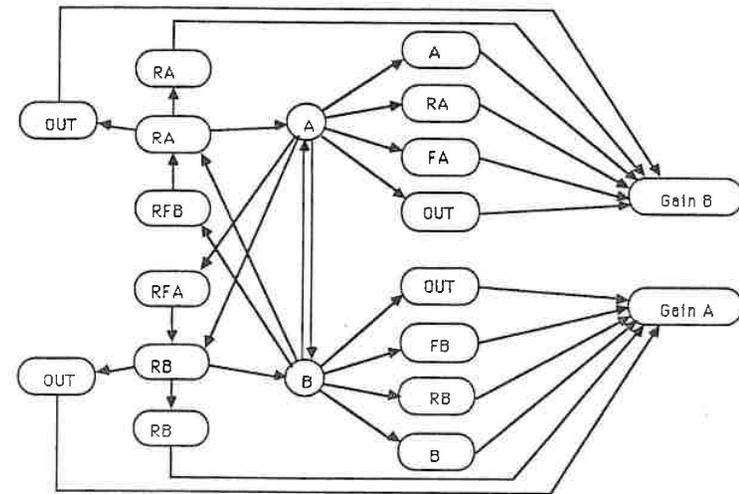
En particulier ces objets doivent posséder un comportement, qui décrit leur évolution dans le temps, puisque le problème traité ici est celui de la perception temps réel. Un modèle simple et classique de comportement peut être linéaire (objets allant à vitesse ou à accélération constante). D'autres modèles d'évolution peuvent inclure des changements discrets qu'on décrit généralement à l'aide de modèles poissonniens de saut de paramètres (voir [Bennis & al], modèle de description du comportement d'avions pilotés). Enfin, ce comportement peut être représenté par des règles de comportement, plus qualitatives ou par une séquence de certains comportements simples. On peut d'ailleurs noter une forte ressemblance entre les descriptions du comportement par un automate ou par des règles.

Les attributs numériques ont donc trois types qualitativement différents de variation correspondant à ces différents modèles d'évolution dynamique: fixes (objet immobile ou de forme apparente constante), variables continuellement (taille ou fréquence variable = mouvement) ou catastrophiques (apparition ou disparition d'un objet, allumage d'un feu rouge). Ceci nous amène à la notion plus vague d'événement, qui peut se définir par un changement brutal d'un de ces attributs, pouvant éventuellement impliquer plusieurs objets.

Dans notre exemple de partie de tennis, nous avons déjà vu plus haut les représentations objets utilisées pour les joueurs, la balle, les trajectoires et les événements. A cause de la syntaxe forte entre les événements, par rapport à une scène de surveillance plus réaliste, nous pouvons décrire l'enchaînement des états possibles de l'échange par un automate représenté ci-dessous. On a omis ici la partie relative au service pour ne considérer que l'échange lui-même, avec par exemple l'initialisation en A. Le jeu, et son résultat final (gain de A ou de B), est ainsi défini par une véritable grammaire associée à l'automate. Celle-ci nous donne la sémantique du jeu. Il s'agit bien sur ici d'un cas particulièrement simple de sémantique, mais qui peut être reliée à tous les systèmes de "scripts" tels qu'ils sont utilisés en intelligence artificielle pour la compréhension du langage naturel [Minsky]. Les étiquettes dans chaque état ne caractérisent pas l'état, mais l'événement qui a lieu dans cet état.

Liste des événements:

- A : A frappe
- B : B frappe
- RA : Rebond côté A
- RB : Rebond côté B
- RS : Rebond dans le rectangle de service
- FA : Balle dans le filet côté A
- FB : Balle dans le filet côté B
- RFA : Rebond sur le filet venant de A
- RFB : Rebond sur le filet venant de B
- OUT : Out



Automate à états finis décrivant un échange de balle

2.2.4 Niveaux de représentation et incertain

Nous savons que les résultats de traitements sont tous plus ou moins entachés d'erreur. Les représentations issues de ces traitements ne peuvent donc être certaines, et sont alors plutôt des densités de probabilités ou des ensembles d'hypothèses. Par exemple, un attribut position d'un objet pourra avoir une distribution gaussienne dont on apprécie la moyenne et la variance.

Nous arrivons alors à un problème-clé pour un système réel, à savoir le traitement de l'incertitude. Les raisonnements et les décisions seront en effet d'autant plus fragiles que les hypothèses sont plus incertaines, cette incertitude pouvant intervenir à plusieurs niveaux. Il semble qu'un "slogan" inspiré de [Dubois & Prade] illustre bien ce problème de validité des décisions:

"Mieux vaut un objet certain et imprécis qu'un objet précis et incertain"

Ici, l'incertitude porte sur l'existence de l'objet en question, tandis que la précision est caractérisée par un domaine de variation pour chaque attribut de cet objet (qui peut aller de n'importe quoi à une donnée précise). Il s'agit donc de favoriser la détection sûre d'objets même mal cernés. Les décisions absurdes et mal fondées, ou fondées sur une fausse interprétation de la scène, sont en effet plus risquées que celles basées sur une interprétation imprécise, dans la mesure où il existe souvent une certaine continuité du risque, fonction des paramètres de l'objet détecté. Cette continuité est plus difficilement vérifiée dans le cas d'une fausse alarme ou d'une fausse classification!

La structure hiérarchisée des représentations, à travers la notion d'objet, entraîne ainsi différents niveaux d'erreurs. Par exemple, si on détecte un camion de vitesse 40 km/h, on peut avoir une erreur sur la vitesse, ou alors une erreur sur l'existence du camion, qui peut être une fausse alarme (auquel cas la vitesse de 40 km/h n'a plus grand sens). L'incertitude à ce niveau est donnée par une probabilité sur l'existence de l'objet. Par exemple:

"il y a 60 % de chance qu'il y ait un camion, et si ce camion existe, et sa vitesse est comprise entre 39 et 42 km/h avec une probabilité de 90%"

De même les erreurs de traitements seront séparées en deux catégories:

- les erreurs qualitatives, qui correspondent à la non détection de l'objet ou du phénomène mesuré, et qui renvoient donc un résultat incohérent (qu'on peut modéliser par une distribution uniforme sur une certaine plage ou par une gaussienne très étalée)
- les erreurs quantitatives ou de précision, souvent modélisées par des distributions gaussiennes, avec une simple approximation aux deux premiers moments (voir chapitre suivant pour une étude plus complète des distributions d'erreur de traitements)

Un traitement peut donc être modélisé par une probabilité de renvoyer un résultat incohérent, par exemple selon une densité de probabilité uniforme, et par une densité de probabilité d'erreur autour de la situation réelle. Grâce à cette description des erreurs suivant deux niveaux, on arrive à une description plus fine et plus réaliste de la distribution des observations qu'avec une classique gaussienne sur un niveau.

Il faut noter que la structure hiérarchisée adoptée nous permet de simplifier cette logique probabiliste sur plusieurs niveaux: si l'hypothèse mère d'un ensemble incertain de valeurs a une probabilité nulle, les probabilités affectées à ce second niveau n'ont plus de signification. Nous verrons en 2.3 que les niveaux permettent de bien structurer le problème de la perception d'un robot multicapteur. Comme on a souvent du mal à évaluer les performances globales de systèmes aussi complexes, on se ramène à étudier les performances des différents traitements et classifications successifs. De ce point de vue également, il vaut mieux que les premiers niveaux soient très sûrs, de manière à ne pas faire écrouler tout l'édifice, comme dans la perception du contexte vue plus haut.

2.3 Architecture du système de perception multicapteur

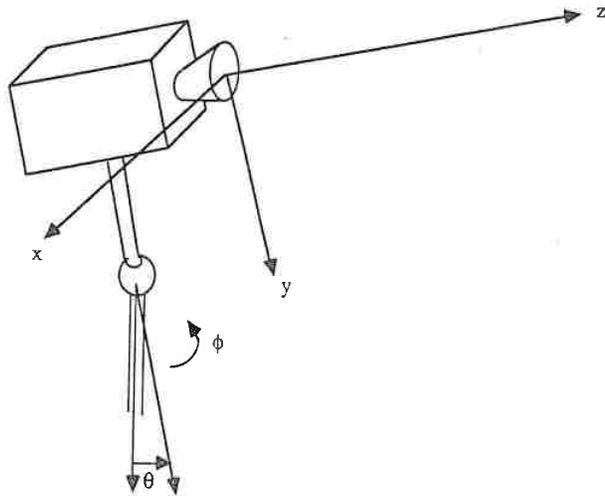
2.3.1 Modèle des capteurs et traitements

Dans un univers robotique, les capteurs et les traitements peuvent être modélisés à deux niveaux: un niveau purement géométrique de génération des images sensorielles à partir de scènes 3D réelles (par des projections...), et d'extraction des primitives de ces images (phénomènes de masquage...). Des exemples de capteurs sont les caméras, infra-rouge ou visible, les microphones, les télémètres et autres dispositifs de mesure de distance. Certains "capteurs" sont en fait déjà des traitements, comme les systèmes calculant la distance par vision stéréo, ou la carte acoustique obtenue par balayage électronique, ou par diagonalisation de la matrice de covariance, à partir d'une antenne de microphones.

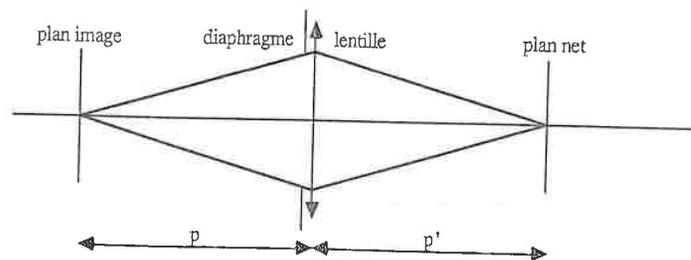
Le modèle géométrique des capteurs est le suivant: l'univers du robot peut être décrit comme un ensemble de sources ponctuelles d'émission de signaux multispectraux (lumière visible, infrarouge, acoustique...) dans un univers géométrique 3D (ou plus si l'on introduit les vitesses). En général, les objets réels sont constitués de plusieurs sources (ou même d'une infinité, pour la vision). Celles-ci constituent donc souvent un niveau perceptif de représentation intermédiaire. Dans la mesure où les capteurs sont considérés comme ponctuels, une part importante de leur description concerne l'étude de la projection géométrique de l'univers tridimensionnel dans l'espace sensoriel des capteurs, volontiers mono ou bidimensionnel.

Rappelons (1.1.1) que le capteur est alors défini par son champ de vision, sa position et son orientation donnée par deux angles. Il apparaît donc comme une **fenêtre d'observation** définie par ses limites géométriques et spectrales. Celle-ci constitue une ressource du système perceptif (dans l'accès aux observations), au même titre que les ressources d'un système informatique, contrôlée par les commandes et les paramétrages capteur (par exemple pour une caméra: orientation, zoom, mise au point...). Par analogie avec la commande d'un bras de robot, on peut ici aussi utiliser des transformateurs de coordonnées pour viser une zone de perception précise (dans [Amat] par exemple, orientation zoom et mise au point d'une caméra pour pointer une zone précise). Il s'agit généralement de transformations simples. Pour le cas particulier de la vision, les problèmes de perspective et de projection des objets peuvent cependant nécessiter des raisonnements assez complexes [Brooks].

Les capteurs sont également caractérisés par leur résolution et leur pouvoir de séparation géométrique et spectral, qui donnent respectivement la finesse des détails observés, et leur aptitude à discriminer deux objets proches. Suivant la précision du capteur, les objets détectés seront situés plus ou moins précisément en site et en azimuth. Les capteurs de vision ont généralement un très bon pouvoir de séparation, tandis que les capteurs acoustiques sont très imprécis en résolution et en séparation.



Le point de coordonnées (x, y, z) est projeté en $(K_x \cdot x/z, K_y \cdot y/z)$ dans l'image. La fenêtre d'observation est délimitée, par la largeur de champ, donc par la focale pour un élément sensible (CCD ou autres) donné.



Position du plan net: $1/p + 1/p' = 1/f$, où f est la focale du système optique.

Lorsque la source n'est pas sur le plan net, l'image est flouée par un filtre correspondant à la transformée de Fourier de l'ouverture du diaphragme. Profondeur de champ pour une tache de largeur inférieure à $2 \cdot z_0$, et pour un diaphragme de rayon ρ :

$$1 - \frac{z_0}{\rho} \leq \frac{p'}{p} \leq 1 + \frac{z_0}{\rho}$$

Exemple de modélisation simple de caméra

Un dernier point caractéristique des capteurs utilisés en robotique, est qu'ils ne sont pas additifs dans la superposition de deux signaux (sauf en acoustique). Le phénomène important dans les lois de projection est alors celui du masquage, qui veut que l'objet le plus éloigné disparaisse derrière l'objet le plus proche se trouvant dans la même direction. Là encore, un raisonnement de type logique perceptive est possible et a été constaté chez l'homme (différents niveaux de développement perceptif chez l'enfant [Piaget], par exemple la notion de permanence de l'objet, même caché derrière un arbre...), mais n'a pas encore été utilisée explicitement. Une version simple de cette logique est un modèle où tous les objets se déplacent dans des plans parallèles entre eux, et perpendiculaires à l'observateur [Millour]. De cette manière, on a un ordre de priorité constant dans le masquage entre les objets.

Les traitements travaillent donc sur ces images sensorielles, dans lesquelles ils essaient de retrouver des primitives particulières à chaque type d'image. Chaque capteur génère en effet celles-ci suivant des logiques de formation et d'évolution spécifiques. On peut prendre pour exemple la détection de contour en image de vision.

Un contour peut être modélisé simplement par une marche bidimensionnelle, avec un bruit gaussien additif dû à la caméra. A partir de ce modèle, on peut étudier en fonction du rapport signal/bruit deux critères: la performance de la détection, exprimée par un taux de faux contours et de contours non détectés, et la précision, exprimée en nombre de pixels d'écart par rapport à la position vraie (voir paragraphe 2.5.1 pour une description plus complète des niveaux d'erreur). On peut de cette manière réaliser des détecteurs de contour optimaux [Modestino][Bouhémy] ou très précis (détection de contour subpixel [Nalva]). Dans ce cas de la détection de contour, on a une bonne modélisation mathématique du phénomène "contour". Mais dans la plupart des autres traitements, le modèle n'est pas aussi rigoureux et la sensibilité au bruit peut donner lieu à de grandes variations dans les résultats (par exemple pour les autres traitements de segmentation).

Les traitements sont donc basés sur un modèle de primitive, qu'ils tentent de retrouver dans les images sensorielles d'une manière plus sûre et plus précise. Le problème de l'incertain, déjà traité en 2.3 pour les représentations d'objet, peut donc également être abordé du point de vue des traitements.

2.3.2 Modèle hiérarchisé

Dans la représentation du système, on trouve non seulement les objets de l'environnement et les capteurs, mais aussi les traitements. En effet, de la même manière que les objets évoluent, que les capteurs peuvent être commandés dans leurs différents réglages, ou même éventuellement tomber en panne, les traitements peuvent faire l'objet d'un choix, d'un paramétrage, ou d'une reconfiguration. On retrouve là encore le problème de contrôle symbolique. Afin de ne pas embarrasser les développements de cette thèse, portant spécifiquement sur les aspects

probabilistes de ce contrôle, nous avons choisi un modèle classique hiérarchique d'architecture du système et d'organisation des traitements.

En général, on repère les traitements et les données par rapport à une échelle comprenant plusieurs niveaux de représentation, soit par ordre de complexité croissante:

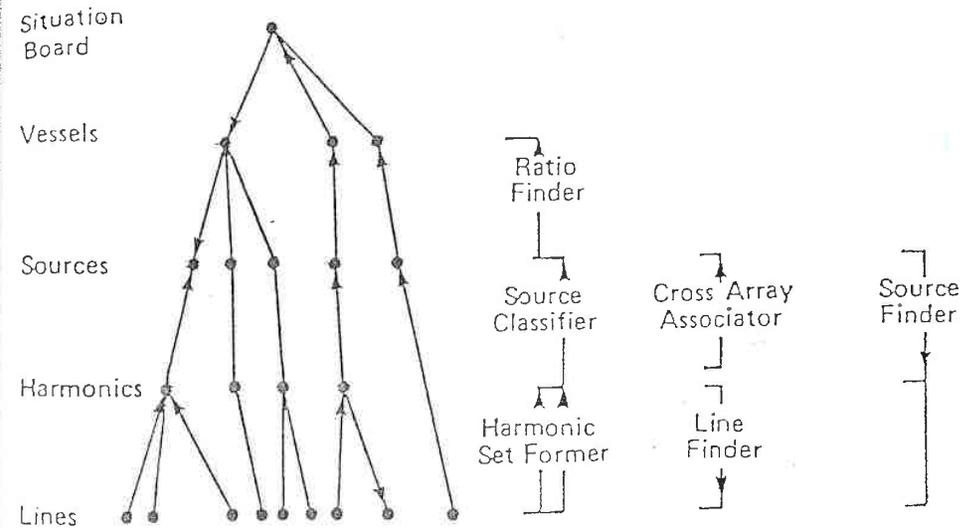
- le bas niveau, qui inclut l'acquisition, et des opérations de filtrage et de recalage des coordonnées
- le niveau lexical, qui correspond à une sorte de vocabulaire des formes élémentaires et à la recherche de primitives
- le niveau syntaxique ou structurel, qui regroupe les descriptions symboliques du niveau précédent en utilisant les contraintes déduites soit d'une grammaire formelle, soit des contraintes d'appartenance ou relationnelles représentées par des arbres et/ou par des graphes. Ce niveau permet la détection des premiers objets ou parties d'objets
- enfin, le niveau sémantique, qui est celui des objets signifiants pour le système de décision, pour la tâche qu'il doit accomplir

Ainsi, dans notre exemple de la partie de tennis, nous pouvons distinguer trois niveaux, le niveau sémantique s'identifiant au niveau syntaxique du fait de la simplicité du modèle et du jeu lui-même:

- le bas niveau correspond aux observations détectées dans la vision et dans l'acoustique: détection de choc, poursuite d'une balle.. Cet espace des observations, résultats de traitement n'est bien sûr pas relié simplement à celui des événements: deux événements peuvent renvoyer le même résultat, et un événement peut générer deux observations différentes de manière probabiliste, ou en fonction de l'état du capteur, ou du contexte d'observation (vision de nuit, bruits parasites troublant l'acoustique..).
- le niveau lexical correspond aux événements eux-mêmes: rebond du côté A ou du côté B, frappe de A.. Ici aussi, un même événement peut correspondre à différents états de la partie, suivant son ordre d'arrivée dans la séquence de l'échange, par exemple RB, rebond de la balle du côté B, signifie deux choses différentes suivant qu'il suit ou qu'il précède une frappe de B. D'autre part, les événements et les trajectoires sont des objets au sens vu en 2.2, qui possèdent des attributs ou caractéristiques spatio-temporels qu'il faut connaître précisément, afin de faire de la prédiction, de tester la cohérence de l'ensemble des trajectoires, et éventuellement d'orienter les caméras et de focaliser les traitements. Ces attributs sont eux aussi précisés par des résultats de traitements.
- enfin, le niveau syntaxique, ou sémantique, est celui des états qui suivent la grammaire spécifiée par l'automate vue précédemment.

Dans les cas réels, le cheminement n'est pas linéaire et l'on peut avoir plusieurs chemins de données possibles, ou un enchevêtrement de plusieurs niveaux, et des sorties et des bouclages transverses (contrôle des actionneurs et des algorithmes - 2.5.3). Ceci est valable dans tous les

cas pour un système multicateur, dont la spécificité est justement la possibilité d'utiliser les redondances des capteurs et des traitements. Pour résoudre ce problème d'enchaînement de traitements dans le domaine de l'analyse de scène, la technique du tableau noir ou "blackboard" a été souvent utilisée, [Nii]. Dans celle-ci, toutes les informations acquises sont mises en commun, et les traitements se déclenchent automatiquement à l'arrivée de nouvelles données. Le modèle est hiérarchisé de manière assez souple, chaque traitement étant repéré par le type ou le niveau de ses données d'entrée et de sortie, et plusieurs traitements pouvant fournir ou traiter le même type d'information (voir figure ci-dessous).



Hierarchisation des représentations et traitements associés dans un modèle du type "blackboard" (Nii)

Kent et Albus ont étudié un modèle hiérarchisé de contrôle en fonction de la perception, chaque capteur ou traitement étant affecté à une des tâches requises par le système, les tâches et les traitements suivant des hiérarchies parallèles [Kent & Albus]. Dans le modèle simple proposé par Henderson [Hansen & al], on voit apparaître pour la première fois la notion de choix dans la perception (choix des capteurs ou des traitements), associé à une organisation hiérarchisée des traitements. Cette hiérarchie correspondant à une représentation des objets perçus sous la forme objet/attribut, sous la forme d'un arbre et/ou, ou celle de toute description syntaxique.

D'une manière concrète, la préoccupation de ce travail est d'élaborer un système qui serve d'interface entre les représentations fournies par les différents traitements et le système de décision qui formule ses demandes dans son propre langage, en terme d'objets et de comportements "signifiants" pour la sémantique globale du système. Il s'agit donc de réaliser une intégration des traitements et de leurs résultats. Dans ce système d'intégration, les capteurs et traitements sont des modules, décrits par leurs caractéristiques. Des systèmes précédents, ont été retenus: l'aspect hiérarchisé, le contrôle local, et la possibilité d'effectuer des choix dans la perception.

2.3.3 Boucles de contrôle hiérarchisées

On aboutit alors à un modèle hiérarchisé de description de la réalité analogue à ce qui se fait généralement dans les programmes d'analyse de scène existants (dans lesquels les niveaux les plus bas utilisent des traitements classiques du genre extraction de contours ou segmentation plus générale). Dans notre système, les aspects originaux sont la présence d'une intégration des informations de diverses sources, qui permet de relâcher les contraintes de chaque traitement, et celle d'un contrôle de la commande capteur, permettant notamment des rebouclages locaux des résultats d'un traitement, comme ce qui se fait par exemple en poursuite.

On peut traiter plus d'hypothèses, puisqu'elles sont résolues dans l'intégration grâce à la redondance entre traitements, dans la mesure où les calculs nécessaires à l'intégration ne sont pas prohibitifs. D'autre part, alors que dans les modèles plus classiques les capteurs sont supposés fixes (excepté par exemple les caméras logées au bout de bras de robot [Rives & Espiau]), nous en prévoyons le contrôle dans notre modèle (orientation, zoom), afin d'optimiser la perception elle-même, et non plus seulement l'action sur l'environnement.

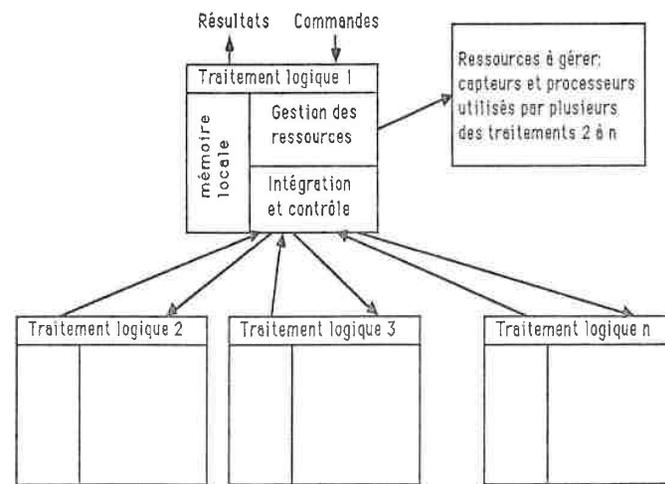
Cette description symbolique, imposée par la complexité d'un système de perception multisensoriel et multitraitement, sera utile dès la conception, comme dans tout système informatique, mais aussi dynamiquement, afin de commander en temps réel les capteurs et les traitements, dans le sens d'une optimisation des fonctions perceptives. Nous expliquons donc notre système à partir d'une description de ses différents éléments constitutifs, séparés en deux catégories: les ressources perceptives, et ce que nous appelons les traitements logiques, qui constituent l'élément de base de notre architecture hiérarchisée.

Les ressources perceptives sont essentiellement constituées des capteurs, dont nous avons vu (1.1.1) qu'ils sont comme des fenêtres d'observation sur ce monde, définies par leurs limites géométriques et spectrales. Un processeur pourra également être une ressource limitative si les traitements ne peuvent tourner en parallèle, cas inverse de la situation où chaque capteur dispose d'un processeur spécialisé. Un autre exemple de ce genre de ressource perceptive est une fenêtre de traitement à l'intérieur d'une image, où un certain traitement peut être fait à la volée, à la vitesse d'arrivée des images [Cassou & Millour], le temps de calcul nécessaire sur toute l'image

étant trop élevé pour le temps réel. Il s'agit dans ce cas d'une ressource intermédiaire entre un capteur et du temps CPU brut, qui peut être appelée ressource traitement.

Le traitement logique (ou capteur logique: [Hansen & al]) est la brique de base de notre architecture, et renferme:

- . un ou plusieurs traitements (simples ou logiques)
- . les commandes de configuration de ces traitements et des capteurs associés
- . un système d'intégration temporelle et multisensorielle (lorsqu'il y a plusieurs sous-traitements). Cette intégration met à jour une mémoire locale du dernier résultat à chaque itération, ce qui permet de réaliser des suivis dans le temps (poursuites)
- . un système de gestion des ressources capteur (en fonction d'un critère d'optimisation de la perception forcément local dans ce modèle hiérarchisé - voir 1.2.3) et éventuellement processeur (choix de l'ordre d'application si les traitements sont séquentiels, ou même répartition temps CPU entre un traitement rapide et un lent comme dans les systèmes informatiques temps réel)



Organisation des traitements logiques

Cette spécification permet d'organiser les données en provenance de plusieurs capteurs ainsi que la hiérarchie relative des traitements. Chaque traitement logique dispose donc d'une couche d'interprétation des commandes de contrôle de haut niveau, et effectue une intégration et un paramétrage correct des traitements.

Le traitement simple proprement dit est un des nombreux prétraitements, extraction de primitives ou traitements capteur disponibles. Il possède plusieurs attributs tels que:

- coût (temps de calcul, vulnérabilité, consommation, etc)
- distribution d'erreurs, et réponses sur les situations possibles (fonction du contexte, conditions météo, ..)

Pour notre robot arbitre de tennis, nous aurons besoin de deux principaux types de perception: visuelle et acoustique. Nous ne précisons pas ici le nombre des capteurs de chaque type, mais dans une simulation plus précise et plus "physique", plusieurs traitements pourraient utiliser le même capteur, ou un seul traitement plusieurs capteurs, ce qui ne manquerait pas de poser des problèmes intéressants d'allocation de la ressource capteur, que nous étudierons d'ailleurs dans les chapitres suivants. Chaque traitement considéré comme un traitement logique doit être décrit numériquement par des probabilités d'observation dépendant de la situation et des réglages des capteurs et traitements. En effet, seul le modèle d'intégration des observations des capteurs nous intéresse; il n'est donc pas nécessaire de simuler exactement la perception, mais plutôt de simuler son comportement et ses erreurs.

Liste des traitements et boucles de base nécessaires pour le robot arbitre:

-Détection et poursuite de la balle:

A chaque instant, on a une probabilité d'accrochage, une probabilité de décrochage, et une probabilité de fausse alarme. Le traitement renvoie les paramètres du segment de parabole suivi, et une incertitude et une confiance dépendant entre autres du temps d'accrochage et du niveau de lumière ambiant et peut-être d'un réglage de zoom. Il s'agit donc d'un asservissement ou d'un bouclage, en plus de traitements simples.

-Poursuite des joueurs:

On suppose dans un premier temps que les joueurs sont suivis en permanence. Par contre, on se donne un grand volume d'incertitude correspondant aux différentes positions possibles de la raquette.

-Détection et classification acoustique des chocs:

Les performances suivent un modèle probabiliste bayésien (on définit les diverses probabilités conditionnelles, également en fonction du bruit ambiant). Eventuellement, on peut préciser une localisation grossière (par exemple le côté du terrain).

-Détection sismique des chocs dans le filet

-Capteurs liés au contexte "météorologique":

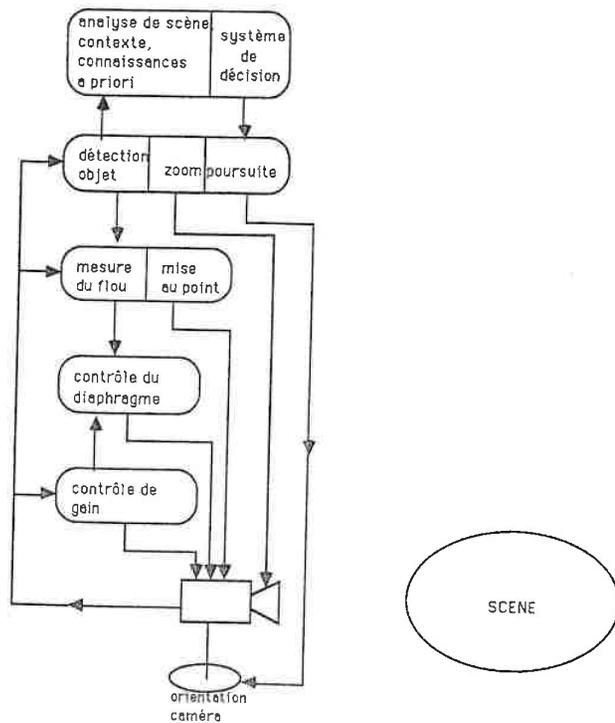
Niveaux de bruit, de lumière.

La notion de traitement logique exploite beaucoup de principes bien connus en génie du logiciel, en particulier ceux d'abstraction, de modularité, et de séparation entre la spécification et l'implémentation. Ces notions permettent d'envisager la reconfiguration dynamique des ressources de perception pour faire soit de la perception adaptée aux situations, soit de la tolérance aux pannes.

D'autre part, cette structure hiérarchisée nous permet de poursuivre différents traitements en parallèle (puisque'ils peuvent être distribués sur des processeurs différents, chacun analysant un aspect de la scène) et d'accumuler et de fusionner les indices sur la scène. Le parallélisme naturel imposé par le multitraitement, le temps réel, et le volume des données, est donc naturellement pris en compte dans cette architecture. Celle-ci doit en effet nécessairement être distribuée et inclure des processeurs de traitement spécialisés afin de réduire le débit des données issues des capteurs, ou plus encore pour permettre d'obtenir des volumes de données à transmettre compatibles avec les contraintes de communication dans le cas de capteurs distribués.

Un aspect non moins important, également dégagé, est l'imbrication de la perception et de l'action. En effet, à l'intérieur de ces traitements logiques, un contrôle local est effectué, entraînant une décentralisation de la décision. Nous avons déjà parlé (voir chapitre 1) de l'obligation du temps réel qui nous impose ces boucles de bas niveau. De plus, ces boucles bas niveau recouvre une notion importante, celle de fonction perceptive. Une fonction perceptive est définie par des éléments de perception associés avec des éléments d'action pour l'accomplissement d'une tâche bien définie (par exemple suivre un objet pour un système de poursuite, ou éviter les chocs pour un robot mobile).

Ce modèle de description hiérarchisé permet en outre de bien cerner les situations imprévues ou extraordinaires, puisque s'il n'y a pas reconnaissance au niveau le plus haut, les couches les plus basses peuvent néanmoins fournir des descriptions valides. En cela, il est également bien adapté au traitement logique ou intelligent de la perception. Comme illustration, la figure ci-dessous montre ce que pourrait être la hiérarchie du contrôle d'un système de poursuite en vision, présentant une certaine analogie avec l'oeil humain.



exemple de contrôle hiérarchisé d'un système de poursuite en vision
(incluant des éléments du projet "plate-forme multicapteur")

Chaque niveau de contrôle est composé d'un module traitement des informations issues du capteur et des niveaux inférieurs et d'un module de commande de réglages particuliers de la caméra. De plus, les niveaux supérieurs peuvent donner des instructions ou prendre le pas sur les boucles plus basses, tout en les laissant tourner toutes seules la plupart du temps.

L'architecture proposée est donc d'inspiration très classique. Elle reflète le parallélisme naturel du multitraitement, logique aussi bien que physique, et inclue la possibilité d'un contrôle et d'une prise de décision locale. L'élément le plus original, outre les techniques utilisées que je décrirai par la suite, est probablement le système d'intégration des résultats et de gestion des capteurs et des ressources perceptives.

Les bouclages sont ainsi réalisés sur différents niveaux. Au niveau le plus haut, se trouve la tâche du système de perception (par exemple: alarme, surveillance de zone). A un niveau inférieur, sont regroupées des tâches de détection d'objets et de classification. Enfin, nous allons détailler ici deux classes de bouclages de base: acquisition des données et poursuite des objets, généralisables bien sûr à n'importe quel couple observable/commande:

1) acquisition de données

A la notion de fenêtre d'observation capteur exposée plus haut, correspond nécessairement une stratégie de balayage ou de scrutation de la scène à observer. Dans un radar de surveillance par exemple, un balayage régulier permet de parcourir systématiquement tout l'horizon. Une autre solution plus évoluée est d'effectuer ce balayage en fonction de ce que l'on a observé. Un exemple particulièrement significatif de ce type de contrôle est l'oeil humain, qui possède une très faible largeur de champ, et qui balais en permanence la scène d'une manière intelligente (par exemple en suivant un contour). Un contrôle de ce type sera étudié au chapitre 4 (4.2). Mais cette solution complexe est due à une limitation matérielle de la ressource "fenêtre capteur" et peut également être remplacée ou simplement améliorée par un capteur à champ plus large (comme l'oreille humaine, ou une caméra grand angle, ou un "fish-eye") avec une résolution et une puissance de traitement suffisante, ou par un ensemble de petits capteurs (comme dans l'oeil de mouche). En dehors du contrôle, un autre problème nécessité par l'acquisition des données en continu est celui de l'intégration de la séquence des perceptions: à quelle niveau celle-ci doit-elle être faite ? Un tel problème est étudié par exemple dans l'établissement d'une carte par un robot mobile.

2) poursuite

Cette classe de bouclage a une certaine parenté avec les balayages "intelligents" vus précédemment, puisqu'elle aussi intègre un asservissement sur les données. Il s'agit cependant plus de suivre un objet, qui doit donc être désigné au départ, éventuellement limité à une région ou un ensemble de points, et qu'il est souhaitable à cet effet d'avoir entièrement dans le champ du ou des capteurs concernés. On peut par exemple envisager des poursuites en azimuth et site, en zoom (pour maximiser l'information sur l'objet), en mise au point, et même en fréquence si l'on dispose d'un filtre ajustable. Le problème se complique dès que l'on cherche à suivre plusieurs objets, puisque des choix sont alors à faire, qui peuvent imposer le recours à plusieurs traitements (détection, classification); la poursuite ne relève plus alors d'une boucle simple.

Au niveau le plus bas apparaît l'alternative classique du contrôle temps réel: acquérir beaucoup d'information à la fois et utiliser des moyens suffisamment puissants pour la traiter, ou en prendre peu (mais de préférence bien choisie) et traiter, et éventuellement intégrer à la volée. Le critère de choix est bien sûr la pression du temps réel, en fonction du but recherché. Les boucles de contrôle de haut niveau seront en général très complexes et très longues. Il est donc

souhaitable pour l'optimisation de la charge des ressources de calcul qu'ils confient le plus de travail possible à des boucles inférieures, quitte à les superviser lors de décrochements ou d'erreurs. Ils auront probablement à effectuer des tâches plus lourdes, en mode asynchrone distinct du temps réel, moins contraint par la vitesse d'arrivée des données (par exemple traitement d'une image hors ligne pour une reconnaissance).

Enfin le dernier aspect dynamique de la perception est la mise à jour permanente d'un tableau décrivant la situation, comme dans les systèmes type blackboard:

- état de la machine: actionneurs, configuration, coûts dynamiques
- connaissances sur la scène: contexte, liste des objets avec attributs

L'intégration des résultats de traitement a ainsi pour résultat la construction et la mise à jour en temps réel de ces représentations, nécessaires au système de décision.

2.3.4 Techniques d'intégration et de contrôle

Un système multicapteur conduit naturellement à une multiplicité des représentations utilisées. Cela vient tout d'abord de la non-superposition des divers espaces sensoriels: une même scène peut être observée par plusieurs caméras, ou même par deux types de capteurs différents, ce qui rend la fusion à bas niveau des objets détectés difficile ou même impossible. De plus, plusieurs traitements peuvent être utilisés; et même si ils détectent le même type d'objet, les attributs peuvent ne pas être précisés de la même manière, ce qui rend une mise en correspondance ou une fusion difficile. Par exemple, dans une même image, un traitement peut renvoyer une indication de détection, tandis qu'un autre ne donnera qu'une indication de distance. Ces problèmes conduisent à essayer d'unifier au maximum les représentations utilisées, en général autour d'indications géométriques: position, attitude, vitesse, ou spectrales (couleur, acoustique), même si les capteurs ne renvoient en général que des indications fragmentaires.

Les représentations issues de l'intégration des résultats des capteurs et traitements sont fournies à un système de décision utilisant un niveau sémantique de description des objets. Nous avons vu au premier chapitre la nécessité d'utiliser ce genre de système, à côté des techniques plus quantitatives d'intégration des résultats de traitements.

La notion de niveau de représentation joue un rôle important dans la structuration du problème complexe de la perception multisensorielle en sous-problèmes. En effet, chaque niveau possède sa propre cohérence interne, qui se traduit par des relations et des contraintes entre les représentations de ce niveau. Dès lors, l'intégration peut être conduite séparément pour chaque niveau, et donc pour chaque traitement logique, en tenant à la fois compte des aspects symboliques et quantitatifs (par exemple traiter une image au niveau de régions 2D liées par des relations de masquage avant de la considérer comme résultat d'une scène 3D).

Nous avons déjà développé un modèle à deux niveaux basé sur la notion de contexte [Lanusse & al], où un arbre de classification de la situation est reconfiguré à la détection de chaque changement de contexte. Dans ce modèle, le premier niveau détecte le contexte, le second classifiant les situations courantes. Il faut noter que le niveau de détection du contexte doit bien sûr être quasi-certain et peu entaché d'erreur, afin d'éviter tout effet avalanche d'erreurs, puisque c'est lui qui pilote le niveau inférieur. D'autre part, comme nous l'avons détaillé en 2.2, nous considérons que les connaissances se séparent en deux classes: les connaissances vagues et les connaissances précises, les premières devant être traitées par un système intelligent, et relevant donc ici de la détection du contexte.

Mais si des techniques intelligentes peuvent être utilisées, si des possibilités différentes peuvent être explorées, celles-ci doivent être ordonnées suivant des critères de coût, incluant des notions de performance, de rapidité, de sûreté, d'utilisation optimale des ressources informatiques ou perceptives, nécessitant un système d'intégration et de contrôle plus quantitatif, qui doit néanmoins être interfacé avec des sources de connaissance plus abstraites, et donc en utilisant des représentations de haut niveau. Pour la partie intégration et contrôle d'un traitement logique, nous proposons un modèle bayésien, dès lors que l'on se trouve dans la classe des modèles précis. D'autres formulations de traitement des incertitudes et de tri des hypothèses sont utilisées, en particulier toutes les théories de combinaison d'évidences.

Les techniques bayésiennes d'intégration ont été choisies en raison de la nature "physique" de la perception. En effet, dans la plupart des cas, on peut trouver un modèle précis markovien d'évolution et d'observation de l'univers, permettant ainsi une estimation récursive de l'état courant (voir chapitre III), alors qu'un tel modèle est en général absent dans les problèmes de fusion d'opinion d'experts. Il s'agit dès lors d'un problème d'automatique des systèmes stochastiques, avec un critère de coût basé sur l'incertitude quant à la situation. On travaille en effet sur un ensemble d'états possibles (ou sur le nuage de probabilité des états, comme dans le modèle bayésien exposé plus loin), que l'on cherche à réduire le plus rapidement possible. Un exemple classique est le modèle linéaire markovien-gaussien utilisé dans les filtres de Kalman.

2.4 Exemples

2.4.1 La partie de tennis

Nous développons ici le problème du robot arbitre de tennis que nous avons évoqué tout au long de ce chapitre.

2.4.1.1 Problèmes soulevés par cette maquette

Nous retrouvons dans cette maquette les problèmes généraux que pose la conception d'un système multicapteur, déjà évoqués au chapitre 1: intégration et contrôle de la perception, plus le problème des événements imprévus.

1) Intégration, utilisation de sources d'informations diverses

Le but du système est de suivre un échange, et éventuellement de faire quelques commentaires sur les coups et sur les joueurs. Le problème est l'interprétation des données partielles, incertaines et imprécises fournies par les capteurs. Pour le résoudre, on cherche à utiliser la redondance entre les divers traitements, ainsi que des connaissances théoriques et expertes sur la trajectoire des balles, sur la structure de la partie et sur les tactiques des joueurs. Chaque source d'information utilise des représentations spécialisées, de type et de niveau différents, qui ne possèdent pas forcément entre elles des liens directs comme "est une partie de" ou "est équivalent à". Des interfaces entre les différentes représentations, possédant chacune leur mémoire de la situation, sont donc nécessaires pour arriver aux descriptions de haut niveau recherchées (attribution du point, échange commenté, étude des performances des joueurs à partir de statistiques).

2) Contrôle, ajustement de la perception en fonction de la situation

Les performances des différents ensembles capteurs+traitements varient en fonction de paramètres externes qui peuvent aller du niveau de la lumière au comportement des objets (par exemple le rebond dépend des qualités de la balle et du terrain) ou à l'état de la connaissance sur la situation (taille de la zone d'imprécision sur la balle, probabilité a priori d'occurrence d'un événement), mais également avec des réglages internes (fenêtre d'observation grâce aux réglages de zoom et d'orientation de la caméra, ajustement de coefficients, ou plus brutalement état de marche du capteur). Le problème est donc d'effectuer à chaque instant le meilleur traitement, soit par ajustement progressif des mêmes algorithmes, soit par reconfiguration logicielle en fonction des performances des capteurs. Là encore, on peut utiliser des méthodes numériques (comme le maximum de vraisemblance, ou des techniques plus complexes de traitement de l'incertain), ou expertes.

3) Imprévus, situations extra-ordinaires

Dans un cas réel, il peut se produire des événements non prévus dans le scénario, comme nous l'avons vu en 2.1.1. Un problème est de faire réagir correctement le système, soit avec une sortie "je ne comprend pas", soit par une description minimale du phénomène ("il y a plus d'une personne du côté A"). Un deuxième événement imprévu à prendre éventuellement en compte est la panne d'un capteur, ou la dégradation de son fonctionnement pour des raisons internes ou externes (météorologiques ou autres). Nous considérerons que ces situations seront décrites sous la forme de situation "hors normes", pour lesquelles il faudrait trouver des indices dans les observations afin de les caractériser. D'autre part, un système de décision intelligent peut être nécessaire afin de traiter tout ce qui est indication vague et peu précise.

2.4.1.2 Intégration des différentes représentations

Le module d'intégration assure la fusion des diverses perceptions, réalise le calcul des erreurs et du maximum de vraisemblance, tout en gérant en temps réel le lancement, la supervision, la focalisation des boucles de bas niveau. Les problèmes à résoudre sont l'évaluation de la confiance accordée aux lignes temporelles, le choix de la meilleure stratégie perceptive possible, et enfin le choix de la stratégie d'intégration en fonction du type des informations disponibles. Le cadre utilisé est du type "blackboard", où divers processus pouvant travailler en parallèle communiquent à travers une mémoire de travail contenant les hypothèses courantes.

Le modèle de la réalité présente une description symbolique du monde perçu sous forme d'objets et de leur comportement. Pour un robot dans un environnement humain, la structuration de cet univers devrait être assez lâche. Une structure reposant sur un modèle d'automate markovien peut être suffisante pour traiter superficiellement un phénomène très complexe, comme dans la reconnaissance de la parole continue [Bahl & al]. Cela pose le problème de la précision nécessaire du modèle de la réalité : une structure lâche, en général hiérarchisée pour des facilités de programmation, suffit-elle toujours, même quand la réalité est beaucoup plus complexe, et quand, par contre, est-il nécessaire d'utiliser des contraintes plus rigides? En fait, plusieurs descriptions de la réalité, de plus en plus contraintes, structurées en couches semblent efficaces.

Les descriptions de base à partir desquels doit travailler ce module sont au nombre de deux: un modèle de la réalité telle qu'elle est observée par le robot décrit dans cette maquette par des objets parcourant des graphes d'état, et un modèle des différents ensembles de traitement vus comme des boîtes noires dont on se donne les probabilités caractéristiques de détection. Ces deux éléments de base correspondent respectivement dans le cas du modèle "markovien caché" adopté ici (voir le chapitre III pour la description de ces méthodes) à la matrice des probabilités de transition entre états $p(S_i, S_j)$ et à celle d'observation conditionnée par les états $p(O_i | S_j)$, mais on pourrait probablement utiliser d'autres types de traitements de l'incertain. On retrouve

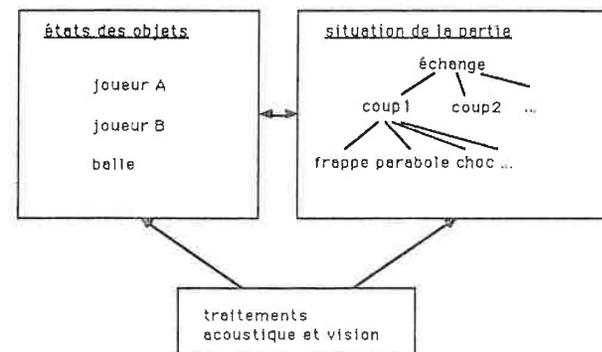
dans le cas du calcul bayésien la probabilité a posteriori d'un état à partir de la probabilité a priori (voir chapitre III), ce qui nous permettra de mêler des informations sur la situation courante (dans $p(S_k)$) et sur l'état de la perception (dans les $p(O_j | S_k)$), toutes informations issues d'un modèle lui-même susceptible de variations par rapport à un comportement normal.

Le modèle de perception est constitué de la description de l'ensemble des boucles de bas niveau sous forme de boîtes noires, ou traitements logiques, décrites par leurs probabilités caractéristiques. Ces traitements de bas niveau génèrent des objets avec une confiance sur leur existence et un volume d'incertitude sur leurs attributs numériques. Elles ont de plus diverses entrées, qui peuvent être des paramètres de réglage, ou une fenêtre d'observation ou d'accrochage initial. D'une manière générale, on donne à chaque boucle de poursuite un volume d'incertitude en entrée sur l'objet concerné, et on en récupère un autre en sortie, en général plus petit, au bout d'un certain temps. Les problèmes qui demeurent sont: l'apprentissage des probabilités caractéristiques, la sortie d'une estimation de confiance et le choix d'une stratégie de focalisation et d'utilisation des ressources.

Ces boîtes noires sont en fait des traitements classiques dans les domaines du traitement du signal ou du traitement d'images. Traditionnellement, les plus simples d'entre eux disposent d'une certaine autonomie décisionnelle pour l'interprétation de l'observation, et réalisent alors une estimation. Cette estimation est par exemple le résultat de l'application d'un simple seuil, et on obtient en fonction de celui-ci une courbe, classique en radar, des variations conjointes des probabilités caractéristiques de non-détection et de fausse alarme. Mais à partir du moment où l'on peut utiliser une redondance avec les résultats d'autres capteurs ou d'autres traitements, ou avec des informations de plus haut niveau, il devient intéressant d'ajuster les paramètres d'un traitement, comme par exemple de faire varier un seuil. En effet, des détections plus douteuses peuvent alors être utilisées, et un certain report de la décision d'estimation peut même être fait vers le niveau supérieur.

Ainsi, un moyen d'éviter de "revenir sur le signal" comme cela peut être fait, par exemple en vision lors de resegmentations descendantes [Hanson & Risemann], consiste à sortir en même temps que la détection un critère de qualité ou de confiance, comme avec l'utilisation de treillis pondérés pour la parole ou de contours plus ou moins sûrs dans les images. L'information fournie est alors plus complexe qu'une simple indication binaire de détection, et tend à fournir la validité de celle-ci. En effet, dans l'intégration bayésienne exposée au chapitre suivant, tout traitement un peu discriminant sur la situation peut être intégré à l'évaluation.

La technique d'intégration est donc à la fois hiérarchisée, temporelle et multisensorielle. Voici une description des différents objets contenus dans la mémoire de travail et de leur organisation:



Contenu de la mémoire de travail

L'organisation verticale correspond à la hiérarchisation des représentations: chaque objet est décomposé en une suite de sous-objets, organisés de manière séquentielle, ou parfois en parallèle (voir vision+son). Le temps introduit une causalité naturelle qui force l'ordre des traitements. De plus, à un instant donné, les traitements ont besoin des informations extraites par le traitement précédent, soit pour un problème crucial de focalisation (dans le cas de la poursuite de balle), soit plus généralement pour l'interprétation. En effet, dans le cas d'un système temps réel, comme à chaque instant il est nécessaire de prendre une décision, et de fournir une interprétation courante de la scène, cela oblige à utiliser toutes les informations disponibles, même si elles sont plus ou moins précises.

Un même module de perception utilisera les résultats des traitements d'une manière différente suivant les objets détectés et la connaissance transmise par les modules précédents. Par exemple, la boucle de poursuite doit être décomposée en traitements élémentaires pour chaque coup d'horloge. On peut envisager une séquence de détection, suivie par une poursuite proprement dite faite de recalage par rapport à un écart, ou même des méthodes du genre filtre de Kalman qui intègrent implicitement l'accumulation de la connaissance sur la balle suivie.

La solution proposée consiste à utiliser une estimation récursive de la situation (voir chapitre III). En effet, on dispose:

- d'un modèle markovien d'évolution de la situation
- d'un modèle de génération des observables

2.4.1.3 Contrôle et focalisation de la perception

Un premier niveau de rebouclage de la perception consiste à faire varier les probabilités a priori de transition et d'observation en fonction des informations contextuelles ou de haut niveau. On optimise ainsi l'utilisation de la redondance des différents traitements en fonction de la qualité des informations acquises. Le deuxième problème de rebouclage qui se pose est celui du contrôle de la perception ou du réglage des traitements. Pour reprendre l'argument exposé plus haut, théoriquement ce problème ne se pose pas, puisque dans un modèle bayésien ou même autre, il suffit d'effectuer en parallèle tous les traitements pour récupérer l'information la plus riche possible. Par exemple, un module de vision de robot n'imitera probablement pas l'oeil humain et présentera un parallélisme plus important, que ce soit au niveau du champ balayé ou à celui des traitements utilisés (une autre possibilité est de multiplier le nombre de caméras).

Néanmoins, pour un système réel, les ressources sont fixes et limitées, et la contrainte du temps réel peut imposer des choix stratégiques:

- ainsi, avec une seule caméra, il faut choisir entre plusieurs directions d'observation (limitation de la ressource capteur)
- pour une caméra donnée et un objet à observer, on peut choisir le zoom qui cadre juste (limitation en résolution)
- avec un traitement limité en temps, il faut choisir une région de l'image sur laquelle se focaliser (limitation en puissance de traitement)
- pour une configuration donnée, on doit limiter le nombre d'objet à traiter grâce à un seuil sur la détection (limitation en temps CPU)

Dans le modèle relativement bien défini de la maquette considérée, il devrait être possible, en fonction des différentes transitions possibles et de leur probabilité, de choisir la stratégie optimale, le modèle de focalisation considéré étant donné. Par exemple, pour la boucle de vision, le choix pourrait être entre plus de précision sur l'objet suivi et le risque de le perdre. On verra un tel exemple de l'optimisation du contrôle de la perception au chapitre 4 (4.2).

2.4.2 Exemple 2: une simulation multicapteur

Pour illustrer les questions évoquées en 2.1.4 et 2.2.4 (incertain, intégration et contrôle), un peu arides, et pour finir de poser le problème, nous décrivons ici un autre exemple.

2.4.2.1 Description

Une simulation a donc été développée sur une machine spécialisée en intelligence artificielle (machine lisp), choisie pour sa puissance de développement et l'interactivité de son interface utilisateur, afin de valider les idées sur les architectures qui permettraient d'intégrer les résultats

de plusieurs modules de traitement travaillant sur plusieurs capteurs. Un exemple d'école bidimensionnel, inspiré d'un jeu vidéo, permet d'isoler plusieurs difficultés de cette intégration, pour lesquelles différentes solutions ont été testées. D'autres scénarios, donnant lieu à des types différents de description de l'univers et des capteurs, ont été étudiés et développés. Ils seront exposés aux chapitres 4 et 5.

Le modèle choisi est un univers bidimensionnel, dans lequel évoluent des objets de formes et d'attributs variés, dont la position et la vitesse (voir figure). Ces objets sont générés aléatoirement d'un côté de l'univers. Ils ont des vitesses variables, mais gardent une trajectoire rectiligne uniforme. Sont ensuite définis des capteurs, également pourvus d'une position, et d'une certaine orientation pour ceux qui sont directifs. Chaque perception à un instant donné pour un capteur est une combinaison en fonction de certaines règles (changement avec la distance, superposition) d'informations issues d'attributs de tous les objets de l'univers. Actuellement deux types en sont implantées: la vision 2D avec les règles correspondantes de perspective et de masquage, et la télémétrie qui renvoie un tableau de distance des objets grâce à un lancer de rayon.

Plusieurs paramètres peuvent être ajustés dans le modèle: la puissance du bruit dans chaque capteur, le taux d'apparition d'objets et leur répartition en vitesses et dans les différents types. L'idée originelle était d'introduire dans un univers simpliste une complexité de bas niveau, et d'évaluer grâce à une mesure unique les performances de divers enchaînements d'algorithmes de traitement et de prise de décision, à imaginer.

La sémantique peut en effet être qualifiée de "primitive" et possède une nature fortement géométrique dépendant uniquement des positions, vitesses et formes des différents objets. Il s'agit en fait d'une version en boucle fermée d'un jeu vidéo simple: des astéroïdes générés par une loi aléatoire menacent un vaisseau spatial (le multicapteur), défini par une zone sensible. Chaque objet qui atteint cette zone tue le système, qui a cependant le droit tous les n coups d'en tuer un à distance pour se défendre. Le but du système est donc de détruire l'objet le plus menaçant au sens d'un certain critère afin de prolonger sa survie.

Le choix de l'astéroïde à abattre dépend d'une décision prise par un algorithme uniquement fonction de la perception issue du télémètre et de la caméra. Deux critères de décision limites servent de référence: ne rien faire pour l'efficacité minimale, et la prédiction de l'objet qui va toucher le premier en accédant directement aux données de la simulation, donc en évitant par court-circuit de la perception. En simulant plusieurs "vies" de ce vaisseau, on arrive à obtenir une mesure simple de la performance d'une séquence de traitements pour ce cas particulier. On se crée alors une base de données qui donne la durée de vie en fonction du type de l'algorithme (qui peut lui-même être modulaire et paramétrable) et des caractéristiques de l'univers.

On se rend compte rapidement que la chaîne de décision vue plus haut (détruire l'objet qui arrive le plus tôt) nécessite des calculs trop sensibles aux incertitudes de traitement, surtout si l'on tient compte du fait que le télémètre, seul moyen d'obtenir des indications de distance, ne renvoie qu'une mesure à chaque instant de la simulation, tandis que la caméra renvoie une image complète. En effet, les calculs à faire sont les suivants: calculer la distance de chaque objet, puis sa vitesse radiale, puis faire un tri de manière à choisir celui pour lequel $\Delta t = r/v$ est le plus faible. A chaque fois, les paramètres des objets doivent donc être connus précisément.

L'optimisation directe du contrôle de ce petit jeu est donc apparue impossible. Il est effectivement apparu à l'expérimentation que les incertitudes sur les attributs position et vitesse étaient trop grandes avec les traitements programmés et les capteurs simulés pour que la décision basée sur ce calcul soit efficace. Par contre, il a été possible de décomposer le problème en sous-tâches, chacune d'entre elle pouvant être étudiée séparément, en prenant en compte une description du type de celle évoquée précédemment, qui précise notamment l'incertitude des résultats. La décomposition du processus de décision trouvée, qui correspond tout à fait aux techniques et représentations d'intelligence artificielle exposée dans ce chapitre, est la suivante:

- 1) éliminer les objets non menaçants, dont la trajectoire n'intersecte pas la zone sensible
- 2) choisir l'objet le plus menaçant à "détruire"

La classification, première partie de cet algorithme peut en effet être réalisée d'une manière efficace et peu sensible aux erreurs et aux incertitudes de traitement. La deuxième partie est par contre réalisée simplement sur un simple critère de distance, et non plus en utilisant la vitesse dans le calcul. Les mesures de performance ont en effet montré que cette nouvelle version était beaucoup plus efficace que la première. On a donc retrouvé expérimentalement le fait (2.2.4) qu'il vaut mieux un traitement sûr et peu précis qu'un traitement précis mais peu sûr.

On se rend donc compte d'une part de l'importance des représentations utilisées en intelligence artificielle, pour décrire les objets et les différents traitements, ne serait-ce que parce que le calcul du contrôle optimal n'est en général pas possible dans un univers peu contrôlé (et dans un univers réel encore plus que dans une simple simulation). Ce sont ces représentations qui vont permettre l'intégration multisensorielle des différentes observations. Cependant, les représentations anthropomorphiques ainsi obtenues ne doivent cependant pas nous leurrer: le calcul d'incertitude, même s'il est approché, est nécessaire pour déterminer la meilleure stratégie (statique pour le choix initial des traitements, puis dynamique pour le contrôle des capteurs et des traitements) parmi l'ensemble des traitements disponibles sur un système multicapteur, et ce sans même prendre en compte les événements imprévus évoqués en 2.1.1. Un algorithme simple et robuste peut donc être plus efficace avec des traitements peu performants qu'un algorithme complexe, comme le calcul de $\Delta t = r/v$, mais sensible aux erreurs. Et il semble d'ailleurs clair qu'un joueur de tennis ne fait pas de calculs précis et explicites de ce genre.

The screenshot shows a simulation interface with two main panes. The top-left pane is titled "Execution" and contains a table of parameters:

Paramètres de l'Initialisation	affichage
taille des triangles: 30	taille de vis: 200.0
taille des cercles: 30	nombre de tirages: 2
vitesse des objets: 20	isoles détectées: 0
vitesse angulaire: 15	groupes détectés: 0
taux de génération: 0.2	fausses païces: 0
pourcentage de triangle: 0	crees: 0
	détruites: 0

Below this table is a list of "Boucles" (loops) such as "BOUCLE-QUI-NE-FAIT-RIEN", "BOUCLE-QUI-TIRE-SUR-LE-PLUS-GROS", etc. The bottom-left pane is titled "Graphics Pane" and contains a list of objects with their coordinates and other attributes. The bottom-right pane is titled "Interaction Pane" and contains a list of actions like "LANCE" and "TIRE".

visualisation de la simulation

2.4.2.2 Problèmes soulevés et résultats

Le système de décision inclus dans ce modèle doit donc a priori connaître la liste des objets présents, ainsi que leurs position et vitesse. En effet, si on raisonne sans tenir compte des problèmes de perception, l'algorithme le plus simple consiste à "détruire" l'objet le plus menaçant, c'est à dire celui qui va arriver dans la zone sensible le plus tôt. On a donc cherché à construire un enchaînement de traitements destiné à reconstruire ces informations à partir des signaux des capteurs simulés. Le critère de performance qui permet de juger et de comparer ces différents algorithmes de décision composés d'une chaîne de traitement est simplement la "durée de vie" moyenne du système, c'est à dire le temps moyen avant l'arrivée d'un objet dans la zone sensible.

CHAPITRE III:

Modèle probabiliste d'évolution de l'univers et des capteurs

Dans ce chapitre, nous détaillons tout d'abord les techniques bayésiennes, ainsi que le principe d'une estimation réursive. Puis nous exposons le modèle de contrôle des systèmes stochastiques imparfaitement observés, avec en particulier la notion importante de "dual control", ainsi que les applications évoluées dans le domaine de la poursuite multicible, dans lesquelles nous commençons à voir apparaître des techniques d'IA lors du développement des arbres de possibilités liés à l'évolution des systèmes discrets, et avec les heuristiques adaptées au domaine. Nous détaillons enfin les particularités du contrôle perceptif, illustrées au chapitre suivant par des exemples d'utilisation de la notion de contexte dans un modèle probabiliste, et de contrôle de fenêtre d'observation capteur, avec en particulier un théorème sur l'allocation optimale d'une ressource dans un modèle stochastique simple.

3.1 Les stratégies bayésiennes

3.1.1 Un jeu contre la nature

Si nous reprenons notre problème de pilotage robotisé d'une voiture abordé au chapitre I ou les exemples vus au chapitre II, le cadre général du problème est celui de la théorie de l'automatique, même si nous l'abordons sous l'angle plus particulier et relativement original de la perception. Il est connu depuis longtemps [Wald][Sworder][Kaufmann 64], et cela apparaît clairement dans ces exemples, que les fondements de cette théorie reposent sur la théorie mathématique des jeux, et plus précisément d'un jeu à somme nulle entre deux joueurs, entre la nature, et le système automatique ou le robot, ou même l'ingénieur concepteur. Le jeu est à somme nulle car un gain d'un joueur correspond forcément à une perte de l'autre.

Introduisons brièvement les éléments de cette théorie: un jeu simple peut être décrit par un tableau rectangulaire décrivant les pertes du joueur A, et donc les gains de B, en fonction des choix de A et B.

	B	1	2
A			
1		-1	3
2		4	-2

tableau décrivant un jeu à somme nulle entre A et B

Chaque joueur choisit donc son coup, qui ici est équivalent à une stratégie puisque les joueurs ne jouent qu'une fois, dans l'ensemble $S_a = S_b = \{1, 2\}$. D'une manière plus générale, un jeu à deux personnes à somme nulle est caractérisé par:

- 1) l'ensemble non-vide S_a des stratégies du joueur A
- 2) l'ensemble non-vide S_b des stratégies du joueur B
- 3) une fonction de coût scalaire H définie sur $S_a \times S_b$

Dans notre cas, A correspond à l'automatisme ou au robot, B à la nature, et H caractérise l'écart à l'objectif ou à la tâche du robot, qui peut être globale ou uniquement perceptive (voir 1.2.3).

Le jeu décrit ci-dessus pose un problème: en fonction du choix de la nature (joueur B), le meilleur coup de A s'inverse complètement. En l'absence totale d'informations a priori sur la stratégie de B, un critère permet de minimiser le coût du pire cas. Il s'agit du critère du minimax, qui s'écrit:

$$\text{choix de A} = \underset{a \in S_a}{\text{Arg min}} \quad \max_{b \in S_b} H(a, b) \quad (3.1)$$

En l'occurrence, le meilleur, ou plutôt le moins mauvais choix de A est donc de jouer 1, puisqu'au pire il ne perdra que 3. Par contre, si par un espionnage quelconque, A peut prévoir le coup de B, alors il est sûr de réaliser un gain dans les deux cas. Une information supplémentaire a donc permis de restreindre l'ensemble S_b des stratégies de la nature, et l'on voit facilement que de cette manière toute nouvelle information ne peut qu'améliorer le critère du minimax, puisque:

$$\text{si } S_b \supset S_b', \quad \max_{b \in S_b} H(a, b) \geq \max_{b \in S_b'} H(a, b)$$

C'est là que nous allons retrouver notre problème de perception: ces informations supplémentaires sont en effet, en plus d'informations a priori donnant un modèle des réactions de la nature en fonction d'un vecteur d'état du système, les résultats d'une ou plusieurs

observations préalables. Bien entendu, la situation n'est pas symétrique, et la nature ne peut observer le système, puisque nous la supposons non intelligente. Cela peut par contre poser un problème dans le cas de lutte contre un adversaire intelligent et hostile.

Pour modéliser à la fois l'incertitude intrinsèque du choix de la nature, qui prend le rôle du joueur B, et des informations sur celle-ci, ainsi que pour élargir les possibilités de stratégie offertes au robot (joueur A), la théorie statistique étend les ensembles S_a et S_b des stratégies simples vues ci-dessus, dites **stratégies pures**, aux ensembles S_a^* et S_b^* des stratégies probabilistes, dites **stratégies mixtes**. Dans l'exemple vu ci-dessus, une telle stratégie est décrite par deux nombres p et q tels que $p+q=1$, le joueur ayant la probabilité p de jouer 1 et q de jouer 2. Si les informations sont donc suffisantes pour décrire la stratégie b^* de la nature de cette manière, la stratégie optimale du joueur A est simplement déterminée par minimisation de l'espérance du coût:

$$\text{choix de A} = \underset{a^* \in S_a^*}{\text{Arg min}} \quad E_{a^*, b^*} (H(a^*, b^*)) \quad (3.2)$$

Il est généralement possible de déterminer une telle stratégie probabiliste pour la nature. Celle-ci est déterminée par le calcul des probabilités conditionnelles en fonction des informations a priori et des observations, d'où le qualificatif de **bayésienne** pour la stratégie déduite du critère (3.2). L'optimalité de cette stratégie bayésienne qu'on en déduit pour le contrôle permet donc d'affirmer que, comme pour le critère (3.1) du minimax, toute nouvelle observation est à prendre en compte dans le calcul des probabilités a priori pour améliorer les performances du système.

Dans le cas où l'on ne peut déterminer les probabilités a priori, il n'y a pas de stratégie uniformément optimale sur les stratégies pures de la nature, et on peut seulement se définir un ensemble de stratégies dites **admissibles**, telles que pour chacune d'entre elles, il n'en existe aucune autre qui soit uniformément meilleure. Par définition, $a^* \in S_a^*$ est admissible si et seulement si:

$$\forall a_1^* \in S_a^*, \quad \forall b \in S_b, \quad H(a^*, b) \leq H(a_1^*, b)$$

Cependant, toute information ou restriction sur les stratégies possibles pour B, donc de S_b , peut qu'amener à l'existence d'une stratégie a_2^* , uniformément meilleure que a_1^* sur S_b' (avec $S_b \supset S_b'$). Le critère du minimax est comme précédemment toujours applicable [Wald]. Néanmoins, lorsqu'on étend le problème à des jeux se déroulant sur plusieurs instants consécutifs, correspondant à des applications automatiques ou robotiques réelles, certains auteurs mettent en doute son intérêt (même s'il est applicable sans problème à une profondeur d'exploration quelconque). En effet, l'utilisation du minimax suppose alors implicitement un comportement intelligent et hostile de l'adversaire, ce qui contredit notre hypothèse précédente sur la nature. D'autres critères ont été proposés pour remédier à ce défaut et intégrer cet

information supplémentaire de la "neutralité" de la nature [Kaufmann 64]: le critère de Laplace, attribuant par défaut des probabilités égales à tous les choix, le critère proposé par Hurwicz introduisant une notion d'utilité.

Dans le cas où la détermination d'une stratégie probabiliste de la nature est possible, grâce aux informations a priori et aux observations antérieures, la stratégie du système est dite bayésienne, car l'intégration de ces informations peut se faire grâce à l'application de la formule de Bayes, la décision nécessitant alors une simple minimisation de l'espérance du coût. On peut également élargir la gamme des modèles probabilistes grâce aux modèles paramétrés, dans lesquels les probabilités a priori sont fonction d'un certain nombre de paramètres, qui permettent eux-mêmes de caractériser la stratégie.

Plusieurs théorèmes importants ont été déterminés dans ce cadre de la théorie des jeux à somme nulle à deux joueurs [Sworder][Kaufmann 64]:

théorème 1 (dont nous ne donnerons pas la démonstration):

Si H est convexe et non bornée, ce qui est en général le cas des fonctions de coût utilisées, alors, pour toute stratégie a^* mixte du joueur A, appartenant à Sa^* , il existe une stratégie pure a dans Sa au moins aussi bonne:

$$\forall a^* \in Sa^*, \exists a \in Sa \text{ telle que } \forall b^* \in Sb^*, H(a, b^*) \leq H(a^*, b^*)$$

Ce théorème prouve que, dans la mesure où la fonction de coût est convexe, on peut limiter l'ensemble des stratégies de contrôle à étudier pour le système aux stratégies pures, et qu'il ne sert donc à rien d'utiliser des stratégies probabilistes pour le robot (joueur A), qui risqueraient en outre de poser des problèmes de mise en oeuvre. L'extension faite plus haut aux stratégies mixtes pour le joueur A est alors inutile.

théorème 2:

Si H est continue, si $a^* \in Sa^*$ est une stratégie bayésienne par rapport à la stratégie $b^* \in Sb^*$ de la nature, et si le support de b^* est l'ensemble des choix de la nature Sb , alors a^* est admissible.

En effet supposons que a^* soit non admissible. Alors:

$$\exists a1^* \in Sa^* \text{ tel que } \forall b \in Sb \quad H(a1^*, b) \leq H(a^*, b) \text{ et } \exists b1 \in Sb \quad H(a1^*, b1) < H(a^*, b1)$$

Comme H est continue sur $Sa \times Sb$, alors sur un ouvert O de Sb contenant $b1$:

$$\exists \eta \quad \forall b \in O \quad H(a1^*, b) < H(a^*, b) - \eta$$

Or le support de b^* contient $b1$, donc:

$$H(a1^*, b^*) - H(a^*, b^*) = \int_b H(a1^*, b) - \int_b H(a^*, b) > \eta \int_b I(O)$$

où $I(O)$ est la fonction indicatrice de l'ouvert O. Il y a alors contradiction avec l'hypothèse selon laquelle a^* est une stratégie bayésienne, minimisant $H(a^*, b^*)$. Ce théorème signifie qu'une stratégie bayésienne du robot, donc optimale pour une stratégie mixte donnée de la nature, est en outre plutôt bonne (admissible = il n'y en a pas d'uniformément meilleure) pour les autres stratégies de la nature, donc d'autres probabilités a priori d'évolution du système.

théorème 3:

Si H est continue et si $a0^* \in Sa^*$ est bayésienne par rapport à la stratégie $b0^* \in Sb^*$, alors $\exists a0 \in Sa$ tel que:

$$H(a0, b0^*) = H(a0^*, b0^*)$$

En effet, on sait que $\forall a \in Sa, H(a0^*, b0^*) \leq H(a, b0^*)$

Or $H(a0^*, b0^*) = \int_a H(a, b0^*)$ car on peut inverser les espérances puisque H est continue.

Donc on doit nécessairement avoir $H(a, b0^*) = H(a0^*, b0^*)$ sur tout le support de $a0^*$

Les théorèmes 2 et 3 montrent que l'on peut restreindre la recherche de stratégie de contrôle aux stratégies bayésiennes pures dans les systèmes "physiques" généralement rencontrés. En fait, le théorème réciproque, que toutes les stratégies admissibles sont bayésiennes, n'est pas toujours vrai mathématiquement. On peut cependant le montrer pour des cas particuliers, par exemple lorsque la taille de l'ensemble Sb des stratégies pures de la nature est finie.

La question de l'intérêt des techniques bayésiennes reste d'actualité, surtout dans les nombreux domaines où le modèle est mal spécifié. Des théories ont été développées, celle des ensembles flous [Zadeh], ou celle des possibilités [Shafer], visant à maîtriser l'incertitude dans ce genre de problème. Mais on observe actuellement un regain des stratégies bayésiennes, avec des arguments théoriques en faveur de son optimalité [Lindley], voire leur utilisation dans des domaines mal formalisés d'expertise humaine, par exemple en diagnostic médical [Spiegelhalter]. Ceci peut être également dû au caractère "naturel" des probabilités pour les utilisateurs, qui seraient ainsi paradoxalement plus adaptées pour une utilisation dans les systèmes experts que d'autres techniques créés pour cela (possibilités..).

Certains auteurs décrivent ainsi des techniques de calcul des probabilités a priori, lorsque les connaissances initiales peuvent être décrites sous forme de contraintes d'égalité ou d'inégalité

entre les probabilités élémentaires: on cherche alors le maximum d'un lagrangien incorporant une entropie et les contraintes supplémentaires [Cheeseman]. D'autres [Spiegelhalter] font remarquer que même avec une valeur approximative de la probabilité estimée, l'utilisation des techniques bayésiennes donnent des résultats "suffisamment bons", outre l'"oubli" naturel des conditions initiales qui apparaît dans les filtres récurrents.

Nous ne prendrons pas position dans ce débat très théorique et mathématique. Néanmoins, comme en témoignent les discussions plus détaillées auxquelles sont consacrées les paragraphes suivants, notre choix d'utiliser ce genre de technique peut être justifié si l'on considère la formule de Bayes appliquée à la perception, où O est l'observation, et (X_j) l'ensemble des états possibles (voir (3.3) et (3.4) plus loin):

$$P(X_i|O) = \frac{P(X_i) P(O|X_i)}{\sum_j P(X_j) P(O|X_j)}$$

Dans le domaine de la perception robotique en effet, on dispose souvent d'un bon modèle théorique ou statistique des capteurs et des traitements, et donc la fonction $P(O|X)$ est connue de manière assez précise. Seule l'initialisation $P(X_i)$ des probabilités est parfois imprécise; or l'effet de cette initialisation devient de plus en plus faible au fur et à mesure que le nombre d'observations s'accroît. Une indication relative de la probabilité a priori d'un événement par rapport aux autres est donc suffisante.

3.1.2 Application de la formule de Bayes à l'intégration d'observation

La détermination des stratégies bayésiennes vue ci-dessus implique le calcul des probabilités a priori des différentes possibilités de la nature. Cela implique bien sûr que l'on puisse préciser cette description de la nature sous la forme de différentes hypothèses. Lorsque ces hypothèses forment un espace discret $\{H_i\}$, on calculera ainsi les différentes probabilités élémentaires $p_i=P(H_i)$. Lorsque ces hypothèses forment un espace continu E, on traitera alors des densités de probabilité sur E: $dP(x_i)=p(x_i) dx$. La formule de Bayes nous permet alors de calculer la probabilité d'un événement après observation, connaissant la probabilité a priori et la probabilité conditionnelle d'observation. Elle est donc idéale pour l'intégration et la fusion des résultats de traitement. Sous la forme qui nous intéresse:

$$P(X|O) = \frac{P(X) P(O|X)}{P(O)} \quad (3.3)$$

où X est l'évènement ou l'état du système, O l'observation, et P(O) un terme de normalisation qui peut être calculé par une somme sur tous les états:

$$P(O) = \sum_i P(O|X_i) P(X_i) \quad (3.4)$$

Dans le cas d'une infinité continue d'états, les formules (3.3) et (3.4) sont toujours applicables,

$P(X)$ devenant une probabilité élémentaire $p(x) dx$ et la somme dans (3.4) devenant une intégrale.

Pour en montrer l'intérêt, nous allons appliquer cette formule à deux cas typiques déjà évoqués au chapitre précédent: l'intégration de l'observation d'attributs dans l'évaluation de l'incertitude associée à un objet, et l'intégration des mesures d'un capteur dont les erreurs sont modélisées sur deux niveaux (voir 2.4).

3.1.3 Exemples

3.1.3.1 Intégration d'observation d'attributs

Supposons d'abord que nous ayons déjà une indication grâce à une détection préalable, ou la connaissance simple, de la probabilité a priori d'existence d'un certain objet X. Cet objet possède deux attributs A et B, pouvant prendre les valeurs respectives a_i et b_j , dont des traitements appropriés renvoient les observations OA et OB. OA et OB sont donc des variables aléatoires dont les lois de probabilité dépendent respectivement de A et de B. La probabilité de X se décompose alors, toujours par connaissance a priori, en $\sum \sum P(X, a_i, b_j)$. L'intégration d'une nouvelle observation, OA, se fait ensuite en appliquant (3.3):

$$\begin{aligned} P((X, a_i, b_j) | OA) &= K P(X, a_i, b_j) P(OA | (X, a_i, b_j)) \\ &= K P(X, a_i, b_j) P(OA | a_i) \end{aligned}$$

et

$$P(\text{non}X | OA) = K P(\text{non}X) P(OA | \text{non}X)$$

avec:

$$\begin{aligned} K &= \frac{P(\text{non}X) P(OA | \text{non}X) + P(X) \sum_i \sum_j P(X, a_i, b_j) P(OA | a_i)}{P(\text{non}X) P(OA | \text{non}X) + P(X) \sum_i P(X, a_i) P(OA | a_i)} \end{aligned}$$

En effet, comme les probabilités sont à deux niveaux, existence de X et valeurs des attributs, l'évènement (X, a_i, b_j) suppose X réalisé, et on doit tenir compte dans les calculs de la probabilité de l'évènement complémentaire nonX (X n'existe pas: $P(\text{non}X)+P(X)=1$), ainsi que de la probabilité conditionnelle d'observation associée (ce n'est pas parce que X n'existe pas que le traitement appliqué à A ne peut pas renvoyer OA). On obtient donc:

$$\begin{aligned}
P(X|OA) &= \sum_{ij} P((X, a_i, b_j) | OA) \\
&= \sum_{ij} K P(X, a_i, b_j) P(OA|a_i) \\
&= K \sum_i P(X, a_i) P(OA|a_i) \\
&= P_a / (P(X) P_a + P(\text{non}X) P(OA|\text{non}X))
\end{aligned}$$

En posant:

$$P_a = \sum_i P(X, a_i) P(OA|a_i)$$

D'autre part, la normalisation est un simple facteur multiplicatif qu'on peut ignorer si l'on conserve les confiances dans tous les événements possibles. L'intégration bayésienne des observations devient alors une simple transformation linéaire, la probabilité de chaque événement X_i étant multiplié par $P(O|X_i)$, et on peut revenir à tout instant aux probabilités classiques par simple normalisation. Si les observations des deux attributs sont indépendantes (i.e. $P(OA, OB|a_i, b_j) = P(OA|a_i) P(OB|b_j)$), l'intégration de celles-ci peut donc se faire successivement dans n'importe quel ordre, ou simultanément, dans la mesure bien sûr où ces observations ont été faites au même moment, ou si l'état n'évolue pas. En effet, compte tenu de la remarque précédente, il suffit d'effectuer les produits sur les probabilités des événements élémentaires (X, a_i, b_j) et $(\text{non}X)$:

$$P((X, a_i, b_j) | OA) | OB) = K_1 K_2 P(X, a_i, b_j) P(OA|a_i) P(OB|b_j)$$

et

$$P(\text{non}X | OA) | OB) = K_1 K_2 P(\text{non}X) P(OA|\text{non}X) P(OB|\text{non}X)$$

où la normalisation $K_1 K_2$ peut maintenant être calculée en une seule fois:

$$\begin{aligned}
&K_1 \cdot K_2 \\
&= P(\text{non}X) P(OA|\text{non}X) P(OB|\text{non}X) + \sum_{ij} P(X, a_i, b_j) P(OA|a_i) P(OB|b_j)
\end{aligned}$$

et du fait du caractère multiplicatif de la procédure, il est clair que l'ordre d'intégration de OA et de OB est indifférent.

3.1.3.2 Modèles gaussiens multimodaux

Lorsque l'on traite de densité de probabilité sur des ensembles d'états continus, un point important est la possibilité de faire des calculs sur ces fonctions. Afin notamment de pouvoir intégrer plusieurs observations successivement, il est souhaitable que la densité de probabilité (dp) a posteriori résultant de (3.3) dans sa version fonctionnelle soit une fonction du même type ou décrite de la même manière que la dp a priori:

$$dp(x|o) = K dp(x) P(o|x) \quad (3.5)$$

C'est notamment le cas lorsque la dp a priori et la dp conditionnelle d'observation sont des gaussiennes en x , puisque le produit de deux gaussiennes dans (3.5) est une gaussienne. Il ne reste donc plus qu'à normaliser avec la constante K pour obtenir la nouvelle dp gaussienne $dp(x|o)$. C'est ce genre de dp qui est utilisé classiquement (souvent implicitement) dans la plupart des calculs d'incertitude de mesures successives, puisqu'il s'agit de la description naturelle lorsqu'on se limite aux deux premiers moments de la loi de probabilité. Mais d'autres possibilités existent, en particulier que ces dp soient une somme de fonctions gaussiennes [Anderson&Moore], ce qui permet de modéliser des erreurs de capteurs plus complexes sur plusieurs niveaux vues au chapitre précédent.

Dans le cas des "gaussiennes contaminées" [Durrant-Whyte] en effet, on peut ainsi modéliser deux niveaux d'erreurs de mesures: les imprécisions, et les "ratages" de la mesure, correspondant à une valeur non significative. Une somme de deux gaussiennes permet ainsi de modéliser ces erreurs de la manière suivante: la mesure a une probabilité pe de renvoyer un résultat complètement faux, donc avec une probabilité à peu près uniforme sur une large bande d'observation, et une probabilité $1-pe$ de renvoyer un résultat exact mais imprécis, modélisé par une gaussienne dont la moyenne est l'état réel, et dont la variance fixe la précision de la mesure. La dp conditionnelle d'observation est donc somme de deux gaussiennes pondérées des coefficients pe et $1-pe$, la première ayant une variance très grande, la deuxième un certain écart-type s .

$$dp(o|x) = pe pu(x) + (1-pe) g(o-x, s)$$

$pu(x)$ étant la dp à grande variance, et $g(o-x, s)$ la loi gaussienne en o de moyenne x et d'écart-type s . Si la dp a priori $p0$ est aussi sous la forme de deux gaussiennes, dont un mode large, alors la densité de probabilité a posteriori, produit normalisé de $dp(o|.)$ et de $p0(.,)$, sera sous la forme d'une somme de fonctions gaussiennes, avec quatre modes. Si $pe \ll 1$, on peut faire l'approximation de fusionner les trois modes faibles dans un seul mode à grande variance, et on est ainsi ramené à une densité de probabilité elle aussi sous forme d'une "gaussienne contaminée".

3.2 Filtrage et contrôle de la perception à partir d'un modèle stochastique markovien

3.2.1 Estimation récursive

Le modèle de contrôle de système stochastique partiellement observé en temps discrétisé utilisé classiquement [Kumar & Varaiya] est le suivant:

$$\begin{aligned}x_{k+1} &= f_k(x_k, u_k, w_k) \\ y_k &= h_k(x_k, v_k)\end{aligned}$$

où x_k est l'état du système, u_k la commande, y_k le vecteur d'observation, v_k et w_k des variables aléatoires gaussiennes non corrélées entre elles et entre deux instants différents, représentant respectivement les bruits de l'observation et de l'évolution du modèle.

Pour le cas particulier d'un système de perception robotique, nous prenons pour x_k le vecteur $(x_{1,k}, x_{2,k})$, où $x_{1,k}$ est l'état du système observé, et $x_{2,k}$ l'état du système d'observation. Par hypothèse, ces deux états évoluent indépendamment (nous nous plaçons dans le cas d'un système de surveillance discret sans action directe sur le milieu environnant), le contrôle u_k influant seulement sur $x_{2,k}$, soit:

$$\begin{aligned}x_{1,k+1} &= f_{1,k}(x_{1,k}, w_{1,k}) \\ x_{2,k+1} &= f_{2,k}(x_{2,k}, u_k, w_{2,k})\end{aligned}$$

Pour des applications robotiques, nous ne nous intéresserons qu'au problème de l'estimation de l'état courant, et non au problème de l'estimation de la séquence la plus vraisemblable à partir d'une séquence d'observations, comme par exemple dans les algorithmes de reconnaissance de la parole. Le problème de l'interprétation de la perception est donc, connaissant la suite $(y_j)_{j=1, \dots, t}$ des observations, de fournir la meilleure estimation x_t de l'état courant, au sens du maximum de vraisemblance, ou au sens du minimum d'une erreur. Si le problème impose, par exemple de mémoriser les n états précédents, on se ramène formellement à ces équations en définissant un nouvel état vectorisant ces n états, de manière à se ramener à un système markovien.

Il s'agit donc d'effectuer un filtrage, ou estimation récursive de l'état courant, c'est à dire le calcul à chaque instant t de x_t à partir de l'observation y_t et d'une prédiction basée sur la fonction d'état calculée à l'instant précédent. Dans le même temps, cette fonction d'état est remise à jour. On verra plus loin qu'en théorie des probabilités cette fonction d'état est la densité de probabilité. D'autres fonctions d'état sont utilisés dans d'autres modèles de traitement de l'incertain (évidences, possibilités).

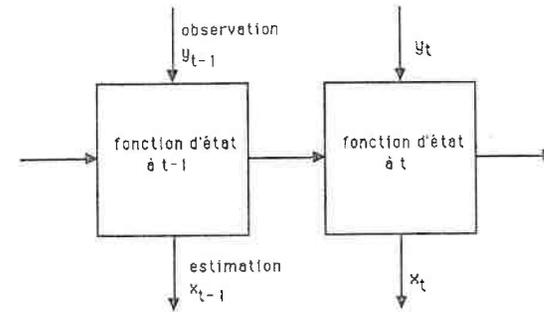


schéma de l'estimation récursive

Dans le cadre des probabilités classiques, la formule de Bayes qui permet d'intégrer à chaque instant les observations représente la manière optimale de traiter les incertitudes [Sworder]. Les réserves qui peuvent être faites, notamment pour le raisonnement grossier dans les systèmes experts, concernent l'impossibilité de calculer les probabilités a priori. Nous avons vu précédemment (en 3.1.1) pourquoi nous avons néanmoins choisi ici ces techniques probabilistes.

Toute l'information utile à l'estimation de x_k , accumulée par la suite d'observations (y_0, y_1, \dots, y_k) est alors condensée dans la fonction densité de probabilité de x_k :

$$pd_{x_k}(x) = P(x_k = x \mid Y_k, Y_{k-1}, \dots, Y_0)$$

Montrons que cette fonction peut être calculée récursivement grâce à la formule de Bayes (3.3) dans les cas discret et continu.

Si nous prenons les notations suivantes:

$pd_{k+1|k}(\cdot)$: fonction densité de probabilité (dp) a priori de l'état au temps $k+1$, prédite à l'instant k

$pd_{k|k}(\cdot)$: dp a posteriori de l'état à l'instant k , après l'observation y_k

$\tau_{x_k}(\cdot \mid x_k)$: dp conditionnelle de l'état à l'instant $k+1$, sachant que l'état à l'instant k est x_k

$ob_k(\cdot \mid x_k)$: dp conditionnelle d'observation à l'instant k , sachant que l'état est x_k

On peut alors substituer dans (3.3) $pd_{k+1|k}(x)$ à $P(X)$ (la probabilité a priori à l'instant $k+1$, dans l'estimation récursive est en effet calculée à partir de la probabilité a posteriori de l'instant

k), et $ob_{k+1}(y_{k+1}|x)$ à $P(y|x)$. Comme la formule est valable pour tous les x de l'espace des états, on obtient une fonction sur celui-ci:

$$Pd_{k+1|k+1}(\cdot) = K_k Pd_{k+1|k}(\cdot) ob_{k+1}(y_{k+1}|\cdot)$$

où K_k est une constante de normalisation:

$$K_k = 1 / \int Pd_{k+1|k}(x) ob_{k+1}(y_{k+1}|x) dx$$

D'autre part:

$$Pd_{k+1|k}(\cdot) = \int tr_x(\cdot|x) Pd_k|x(x) dx \quad (3.6)$$

La formule (3.6) explicite la phase de prédiction qui permet de passer de la probabilité a posteriori à la probabilité a priori de l'instant suivant. Dans le cas discret, les intégrales deviennent naturellement des sommes sur l'ensemble des états.

L'application de ces formules au traitement du signal est justifiée par la signification "physique" de la densité de probabilité conditionnelle d'observation $ob(y|x)$. En effet, le signal peut être décrit comme le résultat d'une transformation mathématique bien connue à partir du modèle de la réalité, avec l'injection de bruit non corrélé. On peut résumer les formules suivantes par une formule de calcul itératif de la dp à l'instant $k+1$:

$$Pd_{k+1|k+1}(\cdot) = F_k(Pd_k|x(\cdot), y_{k+1}) \quad (3.7)$$

où F_k est la fonction du filtrage bayésien effectué sur la dp. La formule (3.7) nous permet de mettre à jour la dp à chaque instant, après l'arrivée de chaque nouvelle observation et permet donc l'intégration de celle-ci dans la dp "mémoire" du système. Bien sur, comme il s'agit d'un calcul dans un espace fonctionnel généralement de dimension infinie, (3.7) n'est utile que si nous disposons d'un algorithme numérique précis et rapide pour la calculer. C'est le cas lorsque le nombre d'états est fini, comme dans les modèles d'automates markoviens cachés [Bahl & al], ou dans le cas linéaire-gaussien, comme dans les filtres de Kalman bien connus, où la dp gaussienne est complètement décrite par son vecteur moyenne et par sa matrice de covariance.

Mais comme nous disposons maintenant de puissants microprocesseurs de traitement du signal, nous pouvons envisager des calculs ou des approximations plus complexes, nous permettant de traiter un éventail de modèles plus large que ceux finis ou linéaires. Sans parler d'une discrétisation directe de (3.7), qui pourrait être irréaliste dans les cas réels, nous donnerons au chapitre suivant l'exemple d'un filtre de Kalman multimodal basé sur une paramétrisation discrète d'un modèle linéaire. Ce filtre utilise la description sous forme de somme de gaussiennes de la densité de probabilité et nous permet d'intégrer des résultats de traitements de reconnaissance des formes, donc dans un schéma intéressant pour un problème robotique.

D'une manière plus générale, on peut, comme on l'a vu plus haut, imaginer des estimations récursives "exotiques" utilisant des densités de probabilité d'état et d'observation telles que l'application de la formule de Bayes conserve la forme de la densité de probabilité de l'état courant, utilisant d'autres fonctions que les gaussiennes. Des modèles intéressants sont ceux mélangeant des descriptions discrètes et continues, permettant d'intégrer des traitements de détection ou de reconnaissance à des mesures plus classiques.

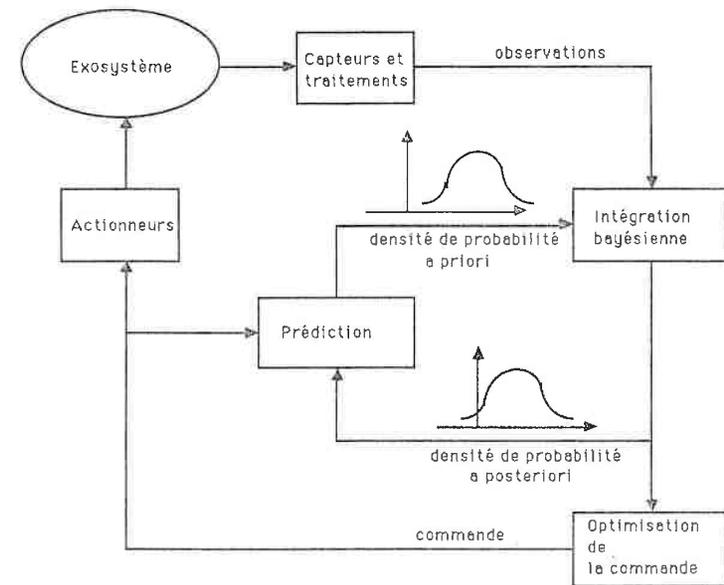


schéma d'une estimation récursive

(les dp gaussiennes ne sont qu'un cas particulier = filtrage de Kalman)

3.2.2 Le contrôle adaptatif, le "dual control"

Comme toute l'information est condensée dans la dp, un contrôle perceptif optimal sera donc seulement fonction de $pd_k(\cdot)$. Par contre, il est clair que l'utilisation de la dp est nécessaire, sauf cas mathématique particulier, et qu'une commande basée sur un simple estimateur sera généralement sous-optimale. Il s'agit d'un contrôle dit séparé:

$$u_k = g_k(pd_k(\cdot))$$

Il est réalisé par minimisation d'une fonction de coût pouvant être défini de différentes manières, à horizon fini ou infini:

$$J_1(g) = E \left(\sum_{l=k}^{k+N} c_l(x_l, u_l) \right)$$

$$J_2(g) = E \left(\lim_{N \rightarrow \infty} \sup_{1/N} \sum_{l=k}^{k+N} c_l(x_l, u_l) \right)$$

$$J_3(g) = E \left(\sum_{l=k}^{\infty} \beta^l c_l(x_l, u_l) \right) \text{ avec } 0 < \beta < 1$$

Dans le cas d'un système de perception pris isolément, on ne connaît pas la tâche accomplie par la boucle robotique, et donc la fonction de coût associée, qui doit alors être pris en compte au niveau global perception + action (voir 1.2.3). Par contre, si la tâche est de la surveillance pure, il n'y a pas d'actions sur l'environnement à optimiser, et on cherche à optimiser la quantité d'information globale sur la scène. On peut alors changer de "point de vue" sur le système, et assimiler la densité de probabilité $pd_k(\cdot)$ à un nouvel état, dont l'équation d'évolution est donnée en (3.7).

La fonction de coût doit dans ce cas faire intervenir $pd_k(\cdot)$, de manière à être maximale quand celle-ci est étalée (l'incertitude est grande) et minimale quand celle-ci est "piquée" (concentrée autour d'un point). Une bonne fonction de coût est l'entropie (ou l'opposée de la quantité d'information), mais d'autres plus simples sont possibles, comme on le verra au chapitre suivant, à propos de l'exemple de contrôle de la fenêtre d'observation.

Le calcul des coûts vu ci-dessus conduit à des équations fonctionnelles stochastiques récursives. La plus simple est celle qui permet de calculer le contrôle u réalisant le minimum $V_0(\pi)$ de J_1 en fonction de la densité de probabilité π , et qui est à la base de la technique appelée programmation dynamique:

$$V_N(\pi) = c_N(\pi)$$

$$V_k(\pi) = \inf_u \left(E \left(c_k(x, u) + V_{k+1}(E_k(\pi, Y_{k+1}, u)) \mid \pi_k = \pi \right) \right)$$

$$u_k = \arg \inf_u \left(E \left(c_k(x, u) + V_{k+1}(E_k(\pi, Y_{k+1}, u)) \mid \pi_k = \pi \right) \right)$$

Ici, $c_k(x, u)$ est le coût à l'instant k de l'état x par rapport à la commande u . $c_N(x)$ est le coût à l'instant N , "horizon" de cet algorithme. Les commandes optimales sont ainsi calculées en remontant le temps de manière récursive depuis $N-1$ jusqu'à t . On détermine donc à l'instant t la série de commandes (u_t, \dots, u_{N-1}) qui minimisent $J(g)$. Cependant comme les calculs sont effectués dans l'espace des densités de probabilité, généralement de dimension infinie, on ne peut en déduire un algorithme que dans des cas particuliers: lorsque le nombre des états est fini, ou dans un modèle gaussien-markovien. Ce type de méthode n'est donc pas applicable en général et on verra au chapitre suivant qu'il faudra utiliser des techniques d'approximation.

Dans le cas d'un critère quadratique d'optimisation de la commande et d'un modèle linéaire d'évolution et d'observation du système, le principe de séparation [Lefèvre] nous dit que l'état à prendre en compte pour le contrôle du système est justement la moyenne de la densité de probabilité gaussienne. Mais dans le cas général, notamment dans celui des systèmes à structure à la fois continue et discrète qui apparaissent dans la poursuite multicible, on ne peut plus séparer l'aspect estimation de l'aspect contrôle. La densité de probabilité doit alors être décrite plus précisément que par ses deux premiers moments, pour être utilisée dans l'optimisation de la commande.

Le contrôle acquiert un aspect double ("dual control" en anglais [Feldbaum]), puisqu'il doit à la fois piloter le système en vue d'un objectif (ici la destruction de la cible), mais aussi contrôler la perception afin d'optimiser à chaque instant la connaissance de la situation. Un exemple de guidage de missile [Birmiwal & Bar-Shalom] illustre bien ce problème du "dual control":

Pour discriminer entre la cible et un leurre, un missile intercepteur doit suivre une direction intermédiaire entre les deux objets, tant qu'il n'a pas identifié l'objectif exact, et qui de plus lui fait voir les objets selon une attitude facilitant la reconnaissance des formes et leur discrimination. Le système doit donc réaliser un compromis entre la minimisation du trajet vers la cible, et l'optimisation de la discrimination des leurres. De même, en détection sous-marine, un changement de direction facilite la levée de l'ambiguïté de distance.

On a donc bien ici un problème de contrôle perceptif, particulièrement lorsque la seule tâche du système est la surveillance, comme nous l'avons détaillé ci-dessus en séparant l'état en deux sous-états: $(x_{1,k}, x_{2,k})$. Il s'agit alors en quelque sorte d'un "dual control", sans la partie contrôle ou rétroaction, ou avec un contrôle limité à l'état $x_{2,k}$ du système observateur. Les techniques d'automatique (automatique des systèmes stochastiques, filtrage non-linéaire) permettent aujourd'hui dans un certain nombre de cas de résoudre ce problème, souvent en faisant des approximations.

Comme nous allons le voir plus loin, ces méthodes peuvent de plus traiter des représentations complexes d'objet, et donc servir dans un système robotique comme interface entre un système de décision intelligent et le résultat des différents traitements de base des signaux capteurs.

3.2.3 le théorème D.A.I. d'allocation des ressources

Nous venons de voir ci-dessus comment contrôler les éléments d'un système de perception. Comme cela a été détaillé au chapitre 2, certains de ces éléments peuvent être vus comme des ressources perceptives, qu'il faut alors contrôler d'une manière continue (fenêtres d'observation capteurs) ou discrète (processeurs de traitement). Le problème du contrôle se ramène alors dans ce cas au sous-problème de l'allocation des ressources dans un système stochastique, sur lequel existent déjà dans la littérature de nombreux articles. Nous allons décrire un théorème important, qui nous donnera des indications pour l'étude au chapitre suivant d'un modèle élémentaire de surveillance avec contrôle d'une fenêtre d'observation.

Ce théorème [Gittins & Jones], [Kumar & Varaiya] se réfère à un modèle assez simple, dit des "bandits à plusieurs bras", par référence à des machines à sous des casinos de Las Vegas. Le problème est d'allouer une ressource (de choisir le bras à actionner) entre plusieurs processus stochastiques, décrits par des n-uplets de la forme:

$$(\Omega_i, C_i, P_i, R_i, \beta_i)$$

Ω_i est l'espace des états du processus i , C_i est l'ensemble des commandes, composé des deux éléments c (pour continue) et a (pour attends), $P_i(x_t, u_t, x_{t+1})$ la distribution de probabilité de transition stochastique de x_t vers x_{t+1} , $R_i(x_{tt})$ le gain associé à x_{tt} , et β_i un facteur de décroissance pondérant la fonction de gain.

On suppose en outre la condition suivante vérifiée:

$$P(x(t), a, x(t+1)) = 1$$

C'est à dire que la commande "a" laisse le processus à qui elle est appliquée en l'état.

Le problème de l'allocation est, étant donné un ensemble de tels processus stochastiques, de choisir la commande u_t désignant à chaque instant t le processus auquel on applique c , les autres recevant a , de manière à maximiser l'espérance de la sommation:

$$\sum_{t=0}^{\infty} \beta_u(t)^t R_u(t)(x_u(t))$$

Ce problème peut être, par exemple, celui de la répartition du temps CPU d'une machine entre différents programmes tournant simultanément, le gain étant alors par exemple opposé au temps passé pour chaque processus (il vaut mieux parler dans ce cas de fonction de coût et traiter le

problème d'une manière symétrique au formalisme adopté ici).

Le théorème (DAI theorem, [Gittins & Jones]) stipule qu'il existe des fonctions $v_i(x)$, de Ω_i dans R , appelées indices d'allocation dynamique (dynamical allocation indices) telles que $u(t)=i$ est optimal pour les états $(x_i(t))$ si et seulement si:

$$v_i(x_i(t)) = \max_j (v_j(x_j(t)))$$

Le théorème fournit de plus une méthode de construction de ces indices, donnée par les équations suivantes:

$$v_i(x_i) = \sup_{1 \leq T \leq \infty} \mathbb{E} \left(\sum_{t=0}^{T-1} \beta_i^t R_i(x_i(t)) \right) / (1 - \mathbb{E}(\beta_i^T)) \quad (3.8)$$

avec $x_i(0)=x_i$, et $x_i(t)$ résultant de l'application de la commande $u(t-1)=i$ sur $x_i(t-1)$. De plus:

$$t^* = \inf_{1 \leq t \leq \infty} \{t : v_i(x_i(t)) < v_i(x_i(0))\}$$

où t^* réalise le supremum en T en (3.8).

Nous ne reprendrons pas la démonstration de ce théorème, qui pourra être trouvée dans [Kumar&Varaiya] chapitre 11. Nous verrons plus loin que ce théorème nous sera utile dans un modèle élémentaire de surveillance par contrôle d'une fenêtre d'observation.

3.2.4 Retour sur les aspects pratiques du contrôle perceptif

Aux algorithmes que nous venons de voir, dont les domaines d'application sont parfois limités, s'ajoutent des indications de type heuristique sur les meilleures méthodes pour le contrôle des ressources perceptives. L'aspect contrôle perceptif est tout d'abord plus particulièrement lié aux fenêtres limitées spatialement et spectralement à travers lesquelles les capteurs observent le monde. Celles-ci peuvent être orientées, où voir leur champ élargi (zoom...). D'une manière générale tout réglage de capteur relève de ce type de contrôle. Le radar à balayage électronique donne un exemple extrême de souplesse, puisqu'on peut avoir un faisceau immatériel, donc non limité par l'inertie mécanique, dont on peut modifier les lobes à volonté. Et comme une seule direction est mesurée à chaque instant, une stratégie de balayage est nécessaire. De même que les autres types de capteurs, ce balayage peut être régulier, ou être fonction de la perception (poursuite, ou balayage de l'oeil humain).

Dans un certain nombre de cas, parmi lesquels ceux vus ci-dessus, des techniques d'automatique utilisant l'aspect "dual control" peuvent réaliser un tel contrôle perceptif. La caractéristique des commandes choisies est qu'elles tendent en moyenne à rendre plus "piquée" la distribution de probabilité de l'état. Le problème de la gestion des ressources est plus vaste, même s'il peut s'inclure dans une même formalisation mathématique. Il se rapproche des techniques d'allocation des ressources dans les systèmes informatiques. Il s'agit en effet de considérer toute la chaîne d'évaluation des hypothèses dans son ensemble de manière à optimiser son fonctionnement, y compris le "moteur bayésien" de réactualisation des probabilités.

Le temps calcul lui-même peut être en effet une ressource du système: si un même processeur a le choix entre plusieurs traitements à effectuer, comment optimiser la performance du système? (par exemple un ordinateur personnel ayant le choix entre différents traitements pour reconnaître un caractère). De même, les nécessités du temps réel peuvent obliger à faire simultanément un traitement rapide, mais grossier, et un traitement plus efficace, mais plus long. En pratique, dans un système temps réel, la configuration est calculée de manière à ce qu'une étape dans la chaîne de traitements soit d'une durée inférieure au temps de variation caractéristique du phénomène observé, quitte à limiter volontairement les données, ou à multiplier les processeurs spécialisés.

Dans certains systèmes, par exemple [Reid], le nombre total d'hypothèses est conservé constant, de manière à saturer à chaque instant les capacités du filtrage, les hypothèses surnuméraires, par exemple celles de poids minimal, ou celles déterminées par un algorithme plus intelligent étant éliminées. On peut ainsi ajuster au mieux la chaîne détection-poursuite (plus la détection fournit d'hypothèses évaluées, meilleure est la performance de la poursuite, mais plus longs sont les temps de calcul). De même, la taille des fenêtres de recherche de la cible est plus ou moins grande suivant l'état du système de poursuite: recherche, cible accrochée plus ou moins précisément.

Enfin, dans l'hypothèse d'intégration de cette perception à un système intelligent, ce type de contrôle se prête parfaitement à un modèle hiérarchique [Kent & Albus]: un bouclage bas niveau peut être fait rapidement par des techniques classiques, tout en étant supervisé par une couche utilisant des connaissances plus complexes, et fonctionnant d'une manière plus éloignée du temps réel. On remarque déjà l'utilisation conjointe de méthodes probabilistes et d'heuristiques (élimination d'hypothèses, technique de fenêtrage..).

3.3 Etude d'une application: les systèmes de poursuite multicible

3.3.1 Description du problème

Dans cette partie, nous nous proposons de montrer, à travers un survol des techniques les plus récentes utilisées en poursuite multicible [Blackman] plus spécifiquement radar, mais éventuellement aussi infra-rouge, comment s'introduisent naturellement l'adaptation aux cibles, le contrôle de ce que nous appelons les ressources perceptives, dès lors que l'on veut prendre en compte l'environnement dans toute sa complexité (multicible, leurres, clutter, fausses alarmes, caractéristiques des cibles).

Un point intéressant est que cette complexification se fait à partir d'un filtrage de Kalman classique, en particulier en rajoutant des composantes à valeurs discrètes dans le vecteur d'état, et en traitant plusieurs objets simultanément. A partir du filtrage de base, on arrive à traiter des arbres d'hypothèses de manière à prendre en compte toutes les ambiguïtés.

Les techniques d'optimisation permettent donc maintenant le contrôle de systèmes complexes, incluant des représentations de type objet, avec des composantes continues et d'autres discrètes. Les systèmes de perception robotique peuvent ainsi utiliser ce genre de méthodes probabilistes comme moyen de traitement de l'incertitude et comme interface entre les résultats des traitements et un système de décision travaillant au niveau sémantique. D'autant plus que ces modèles bayésiens, longtemps délaissés dans les systèmes experts au profit des théories de l'évidence ou des possibilités, connaissent actuellement un regain de faveur [Spiegelhalter].

D'une manière générale, on peut distinguer deux tâches de poursuite: celle destinée à la conduite d'un missile ou d'un canon, et celle de surveillance de la situation, qui doit détecter et suivre les pistes des différents objets présents dans l'environnement. Ces tâches prennent également un aspect différent, plus urgent, en présence d'éléments hostiles pouvant concrétiser une menace (comme dans la détection d'un missile attaquant un navire). Ces différents types de tâches se traduisent par des fonctions de coût différentes, d'horizon fini ou infini, de coûts élémentaires additifs, ou avec des coûts infinis sur les feuilles critiques de l'arbre des possibilités (par exemple en cas de destruction de la cible par le missile).

Dans le premier cas de la conduite de tir, l'aspect automatique du problème est clair, puisqu'il s'agit de contrôler la trajectoire du missile ou des obus. Dans le deuxième cas, il n'y a pas à proprement parler de commande, hormis celle plus immatérielle d'un faisceau radar par exemple. Néanmoins, comme on l'a déjà vu, à partir du moment où l'état réel du système est sujet à ambiguïté, des choix sont à faire pour le système d'observation, dans le paramétrage des différents traitements, dans l'allocation des ressources, qui modifient les observations qui pourront être faites. Ceci constitue ce que nous avons appelé le problème du contrôle perceptif, déjà abordé dans la partie précédente avec le "dual control".

Classiquement, la méthode utilisée pour poursuivre une unique cible est d'effectuer un filtrage de Kalman: à partir d'un modèle markovien linéaire du système, incluant des bruits gaussiens d'observation et d'évolution, on décrit la densité de probabilité de l'état de la cible par une gaussienne. La moyenne et la matrice de covariance de celle-ci peuvent alors être calculées récursivement à chaque instant par la formule de Bayes, en prenant en compte la nouvelle observation.

3.2.2 Complexification de l'univers et adaptation aux cibles

Le fait d'améliorer le modèle de l'univers pour coller le plus possible à la réalité et optimiser les performances des systèmes de poursuite va nous éloigner du modèle simple de Kalman, jusqu'à éventuellement prendre en compte des techniques de programmation évoluée (intelligence artificielle). En effet, pour s'adapter au comportement des différentes cibles et aux erreurs d'observation, on est obligé d'intégrer les phénomènes suivants:

- dynamique ou cinématique de la cible: changement brutal de l'accélération, de la direction, manoeuvres
- différentes hypothèses possibles pour le suivi d'une piste, soit à cause des fausses alarmes, soit à cause d'une mauvaise association de la piste avec une autre cible.
- différents types de cibles, ayant des caractéristiques et des modèles d'évolution différents (leurre, Boeing 747, Mirage), chacun étant doté d'attributs spécifiques, discrets (type du radar embarqué) ou continus (caractéristiques de la surface équivalente)
- traitements supplémentaires permettant l'observation de ces caractéristiques, discrètes ou continues.

Ces différents points peuvent être inclus dans un nouveau modèle markovien, qui garde pour base le modèle linéaire classique en y ajoutant une structure discrète dans les caractéristiques des objets et dans les modèles d'évolution et de perception.

Les manoeuvres peuvent être prises en compte dans un filtre non-linéaire généralisé sous forme de sauts poissonniens dans les variables d'état de la cible. Une manière plus simple de les traiter est d'envisager à chaque instant les deux hypothèses: comportement normal ou manoeuvre, ce qui revient à chercher la cible dans une fenêtre d'observation (on retrouve dans ce cas particulier cette notion définie dans les deux premiers chapitres) plus ou moins grande. On obtient alors un embranchement dans l'arbre des possibilités, que l'on va poursuivre dans le temps jusqu'à ce que la probabilité de l'une des branches deviennent négligeable.

De la même manière, les fausses alarmes ou les mauvaises associations de cible à une piste génèrent des embranchements qui multiplient le nombre d'hypothèses considérées. Il faut noter la nécessité de procéder à un apprentissage de la probabilité de fausses alarmes, qui dépend de

la localisation géographique et varie donc suivant les zones de la scène, et de la météo.

La présence de plusieurs cibles nous conduit à chaque instant à une combinatoire d'associations entre les anciennes pistes et la nouvelle position des objets. La technique du fenêtrage (ou "gating") permet de limiter le nombre de ces associations. D'autre part, l'espace des hypothèses à considérer n'est plus alors la présence ou l'absence de chaque cible, mais plutôt l'espace conjoint de réalisation des hypothèses. Par exemple, si deux objets peuvent être associés indifféremment à deux pistes, seules deux hypothèses combinées sont à considérer:

H1: Ob1 associé à P1
Ob 2 associé à P2

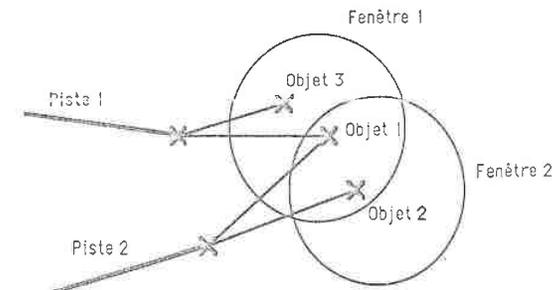
H2: Ob1 associé à P2
Ob2 associé à P1

Les autres hypothèses, par exemple

H: Ob1 associé à P1
Ob2 associé à P1

étant logiquement improbables, puisqu'impliquant la disparition d'une piste et la "fission" d'une autre (bien que cela puisse correspondre, à un autre niveau, à la séparation d'un groupe).

Le calcul des probabilités résultantes s'en trouve compliqué, mais on élimine de la sorte des branches inutiles. En général, ce calcul n'est pas fait pour les groupes d'avions ($n > 3$). Ceux-ci sont plutôt considérés comme des objets à part entière (identique au type "groupe" décrit au chapitre 2), caractérisés par leur position et par le nombre d'avions. La mesure de ces attributs requiert des traitements complémentaires. On trouve là un autre exemple de représentation complexe.



Association des pistes avec les nouveaux objets détectés

Les embranchements d'hypothèses et les caractéristiques des objets s'intègrent parfaitement dans le modèle markovien avec la structure continue linéaire. Si on applique la formule de Bayes à un tel modèle, où les parties discrètes et continues sont indépendantes:

$$P(X_i | Ob) = P(X_i) P(Ob | X_i) / P(Ob) \\ = K P(x_i) P(C_i) P(y | x_i) P(OC | C_i)$$

où K est une constante de normalisation sur tous les états X_i , $X_i=(x_i, C_i)$, $Ob=(y, OC)$, x_i et y étant respectivement l'état et l'observable continus du modèle linéaire, C_i et OC les parties discrètes. On suppose ici que les observations et les probabilités a priori discrètes et continues sont indépendantes (i.e. $P(Ob|X_i)=P(OC|C_i).P(y|x_i)$ et $P(x_i,C_i)=P(x_i).P(C_i)$). Si la distribution de probabilité a priori $P(x_i)$ est gaussienne, $P(x_i) P(y|x_i)$ sera aussi une gaussienne (mais non normalisée) dont les moyenne et matrice de covariance sont données par les formules d'itération usuelles du filtre de Kalman. Par contre, la normalisation devra être calculée sur tous les états possibles, en utilisant une sommation pour les composantes discrètes et une intégrale pour les continues:

$$K = \sum_i^n P(C_i) P(OC | C_i) \int P(x_i) P(y | x_i) dx_i$$

Ces deux structures continue et discrète s'intègrent donc bien dans le cadre markovien. Et c'est là que l'on peut voir la possibilité, voire la nécessité d'utiliser une couche logicielle plus proche du niveau sémantique. En effet, ces attributs et ces choix discrets sont traités d'une manière probabiliste, donc sans autre référence que statistique à leur signification. On voit l'intérêt d'un système qui pourrait attribuer un sens et utiliser la présence de tel type d'avion, tel type de manoeuvre dans telle situation.. En intégrant les informations continues et discrètes issues de divers traitements, et en permettant le contrôle perceptif, ces techniques probabilistes fournissent une interface entre le bas niveau et un système de décision basé sur des méthodes de programmation évoluées. Cette liaison pourrait être réalisée simplement en générant à partir des données fournies par les méthodes logiques et discrètes les probabilités a priori, qui sont alors intégrées à l'estimation récurrente. On trouve en effet dans [Cheesemann], déjà évoqué, une méthode pour le calcul de ces probabilités, à partir d'informations logiques sous forme d'implications, ou de relations de dépendances, en utilisant le principe du maximum d'entropie.

3.4 Conclusion sur l'utilisation des modèles probabilistes

Nous avons exposé dans ce chapitre l'intérêt des modèles probabilistes, qui permettent l'intégration de plusieurs observations d'origines identiques ou diverses, ainsi que l'optimisation du contrôle des capteurs. Ces techniques s'interfaçent également aisément avec des représentations objet complexes, et des informations a priori sur la scène, ou des indications heuristiques sur la manière d'utiliser les ressources capteurs ou processeur. On arrive ainsi à un ensemble d'outils permettant de traiter ce problème du contrôle de la perception dans le cas le plus courant, celui des fenêtres d'observation capteur comme ressources limitatives du système. Enfin, l'attribution des probabilités aux différentes hypothèses sur la scène observée permet une prise de décision tenant compte du bruit d'observation, et des différentes erreurs de perception.

CHAPITRE IV

EXEMPLES D'INTEGRATION ET DE CONTROLE DE LA PERCEPTION

4.1 Un exemple d'utilisation de la notion de contexte: extension du filtre de Kalman

4.1.1 Représentation du contexte sous forme de paramétrisation discrète

Nous avons introduit la notion de contexte d'une manière générale au chapitre II: un contexte caractérise un ensemble de situations dans lesquelles la réaction du robot est déterminée par une même structure courante de configuration des traitements et de la décision. De là le modèle à deux niveaux proposé par Alain Lanusse [Lanusse] [Lanusse&al]: un premier niveau détecte les changements de contexte à partir d'indices particuliers de la situation et des observations courantes, et configure en conséquence un second niveau de traitement et de décision des situations courantes. On retombe donc sur un schéma multi-modèles, le changement de modèle étant effectué sur détection du nouveau contexte:

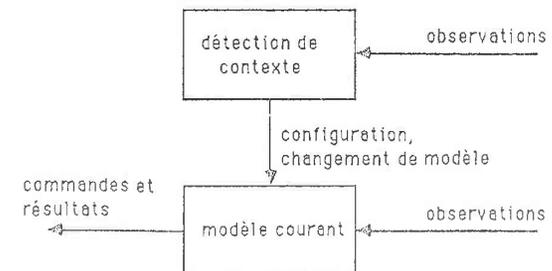


Schéma à deux niveaux: détection de contexte et modèle courant

Nous allons donc dans cet exemple représenter le contexte comme une paramétrisation pouvant prendre un nombre fini de valeurs, donc permettant de faire un choix dans un ensemble de modèles courants de la réalité. Une tendance de ces dernières années en automatique a été d'intégrer de tels aspects à l'intérieur de modèles de commande linéaire classiques. Cette intégration a été réalisée sous la forme d'une paramétrisation discrète avec sauts probabilistes poissonniens entre les différentes valeurs du paramètre dans [Caines & Chen] et [Willisky & Jones].

De tels sauts poissonniens sont souvent utilisés pour décrire des transitions brutales et aléatoires dans le comportement d'un système. On n'effectue plus alors de changement de modèle comme dans la figure ci-dessus, mais la prise en compte des probabilités de chaque modèle, qui sont évaluées à chaque instant en parallèle dans l'estimation récursive. La décision de commande, ou autre, doit alors intégrer ces probabilités. Un exemple intéressant d'utilisation de telles techniques est le contrôle d'un échangeur thermique solaire, dans lequel un capteur entaché d'erreurs permet un ajustement optimal du flux du liquide caloporteur suivant la présence ou l'absence de nuages, qui provoquent des variations brutales de l'éclairement du panneau [Willisky & Jones]. On trouve là une illustration frappante d'une prise en compte d'un phénomène extérieur et non contrôlé à l'intérieur d'un modèle de type automatique, grâce justement à une modélisation poissonnienne des transitions.

La disponibilité de microprocesseurs de traitement du signal rapides et puissants nous permet d'utiliser de tels modèles stochastiques complexes. Or nous venons de voir au chapitre III que les modèles markoviens pouvaient intégrer des structures mixtes continues et discrètes. Nous allons donc présenter dans cet exemple l'introduction de la notion de contexte sous forme de paramétrisation de modèles linéaires-markoviens classiques dans un temps discrétisé.

Dans les techniques de traitement du signal pour la robotique, les filtres de Kalman sont en effet souvent inadéquats, bien que très efficaces lorsqu'ils peuvent être appliqués. Et cela est en partie dû aux aspects discrets de la perception, à la fois dans les états et dans les observations (résultats de reconnaissance des formes...). Une telle extension du filtrage de Kalman classique à partir d'un ensemble discret de contexte pourrait donc avoir une application intéressante en robotique. On bâtit ainsi un exemple relativement simple de l'intégration de représentations du chapitre II avec des modèles probabilistes d'estimation récursive du chapitre III.

Dans ces modèles, la densité de probabilité (dp) dans l'espace des états est mise à jour récursivement à chaque instant grâce à la formule de Bayes, en fonction de la dp a priori et de la dp conditionnelle de l'observation. Par le biais d'une représentation numérique de dimension finie de la dp, nous pouvons utiliser cette formule pour intégrer toutes les observations faites sur le système. Cela est le cas dans les modèles d'automates markoviens à états finis, ou dans les filtres de Kalman, où les dp gaussiennes sont décrites par leur vecteur moyenne et leur matrice de covariance.

Les distributions de probabilité sous forme de somme de gaussiennes satisfont au critère de conservation de cette forme par l'application de la formule de Bayes, et peuvent donc être décrites par un nombre fini de composantes dans une estimation récursive (voir 3.2.1 et 3.1.3.2). Nous nous contenterons donc ici de décrire une extension du filtre de Kalman basée sur ce type de dp, qui, bien que relativement simple, a le mérite d'intégrer les aspects discrets de la perception, et de se prêter à l'utilisation d'heuristiques comme on l'a vu dans l'exemple de la poursuite multicible au chapitre III. Cette description gaussienne multimodale de la dp a déjà été utilisée en traitement du signal [Anderson & Moore] et en traitement de l'incertitude en robotique ("gaussiennes contaminées" [Durrant-Whyte]).

Dans une forme simplifiée du modèle, la paramétrisation peut être utilisée sans sauts entre les états discrets. Dans ce cas, le filtre réalisera simultanément l'estimation du système et l'identification parmi un ensemble fini de modèles linéaire-gaussiens. A chaque instant, il fournit les probabilités de toutes les hypothèses, chacune correspondant à un mode gaussien, et les moyenne et variance des valeurs numériques de chaque hypothèse. On retrouve donc dans ce modèle les niveaux d'incertitude exposés au chapitre II (en 2.4).

Dans sa forme complète, il permet une bonne estimation d'un modèle avec des paramètres discrets dans la scène ou dans les capteurs (on a vu au chapitre II que les états des capteurs sont inclus dans le système), et des observations discrètes, et même avec des sauts entre les états. On peut donc modéliser de cette manière des états discrets tels que comportement normal / panne de capteur, ou présence / absence d'un obstacle. L'observation discrète peut être par exemple le résultat d'un algorithme de reconnaissance des formes appliqués aux signaux capteurs. De cette manière, la notion de contexte peut être intégrée dans les calculs courants d'estimation et de contrôle de la perception.

4.1.2 Le modèle

Dans notre modèle, l'état et l'observation ont tous deux une partie discrète et une autre continue. La continue suit un modèle markovien linéaire-gaussien classique, paramétrisé par l'état discret, qui suit les transitions d'un automate probabiliste. La dp est la somme de fonctions gaussiennes non normalisées, chacune étant un mode de la dp. A chaque incrément de temps, un mode donne naissance à un ou plusieurs modes, d'une manière correspondant aux transitions possibles à partir de l'état de l'automate caractéristique de ce mode père. Les formules d'itération des caractéristiques des modes (intégrale, moyenne, matrice de covariance), dérivent directement de celles du filtre de Kalman classique et de l'intégration des observations discrètes.

Nous ne sommes pas intéressé ici par un simple estimateur de l'état à chaque instant, mais par un calcul de la forme de toute la dp, de la même manière que dans le filtre de Kalman classique

nous calculons à la fois la moyenne et la matrice de covariance, puisque c'est la dp qui condense toutes les informations obtenues. C'est encore plus vrai avec cette description multimodale, puisque l'estimée de l'état au sens des moindres carrés n'a plus de lien direct avec la représentation de la dp. A partir du moment où existent différents modes dans la dp, le principe de séparation n'est plus valide, et une ambiguïté sur l'estimation apparaît.

Cela signifie que nous avons une liste d'hypothèses avec leur probabilité associée, plutôt qu'une seule estimée. De cette manière la perte des informations obtenues dans le passé est minimale. Qu'il s'agisse d'un coût quadratique ou non, l'estimation requise par le problème ou la tâche de contrôle peut être ajoutée après le calcul de la dp. Par contre la combinatoire introduite complique l'optimisation d'une commande.

Considérons le modèle markovien mixte discret et linéaire-gaussien suivant:

$$\begin{aligned} x_{k+1} &= A_k(q_k) x_k + G_k(q_k) w_k \\ y_k &= C_k(q_k) x_k + H_k(q_k) v_k \end{aligned} \quad (4.1)$$

et si $q_k = r$:

$$\begin{aligned} q_{k+1} &= q \text{ avec la probabilité } p_{t_k}(q, r) \\ o_k &= ob \text{ avec la probabilité } p_{o_k}(ob, r) \end{aligned}$$

où (x_k, q_k) est l'état à l'instant k , (y_k, o_k) le vecteur d'observation à l'instant k , $q_k \in S$, $x_k \in R^n$, $o_k \in O$, $y_k \in R^p$, S et O sont des ensembles finis, v_k et w_k sont des variables aléatoires gaussiennes multidimensionnelles non corrélées entre elles et dans le temps représentant respectivement les bruits d'observation et d'évolution, et A_k, H_k, C_k, G_k sont des matrices réelles de la taille appropriée paramétrisées par q_k . L'évolution de q_k est donc indépendante de x , tandis que l'inverse n'est pas vrai. v_k et w_k sont indépendantes et ont pour distributions respectives $N(0, R)$ et $N(0, Q)$.

q_k , le paramètre de ce modèle linéaire-gaussien, suit un automate markovien caché, c'est à dire un automate probabiliste dont les états ne sont pas connus directement, mais à travers des symboles o_k émis de manière probabiliste à chaque état. Il est caractérisé par la matrice des probabilités de transition:

$$p_{t_k}(r, q) = P(q_{k+1}=r | q_k=q),$$

et la matrice des probabilités d'observation:

$$p_{o_k}(o, q) = P(o_k=o | q_k=q).$$

Les matrices du modèle linéaire-gaussien et de l'automate probabiliste peuvent changer avec le temps. q_k ne dépend donc que de q_{k-1} , et non de x_{k-1} . Par contre, d'après les formules vues plus haut, x_k dépend de q_{k-1} .

On fait l'hypothèse que la dp a priori est une somme de fonctions gaussiennes dans les dimensions continues et non-nulles pour un seul état q_n :

$$pd(q, x) = \sum_n \partial(q, q_n) A_n N(x, x_n, \Sigma x_n) \quad (4.2)$$

avec ∂ indice de Kronecker, A_n étant un coefficient réel, et $N(x, x_n, \Sigma x_n)$ étant la gaussienne normale en x de moyenne x_n et de matrice de covariance Σx_n . La dp est donc définie comme une somme de modes gaussiens, chacun étant caractérisé par $(q_n, A_n, x_n, \Sigma x_n)$.

4.1.3 La densité de probabilité multimodale et les formules d'itération du filtre

Par la suite, nous démontrons que si la dp a priori est de la forme (4.2), avec le modèle d'évolution spécifié en (4.1), alors la dp a posteriori résultant de l'application de la formule (3.7) issue de Bayes est aussi de la forme (4.2). Un tel résultat est classique pour les modèles linéaire-gaussiens [Anderson & Moore]. L'extension à un modèle mixte discret et linéaire-gaussien est directe.

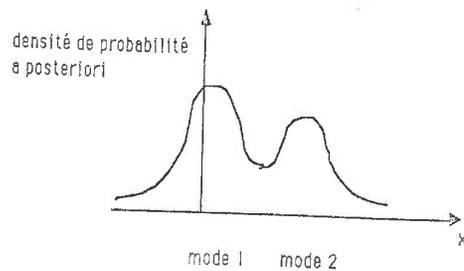
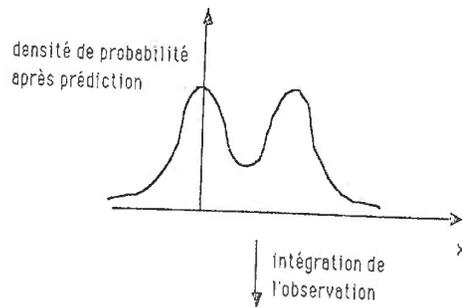
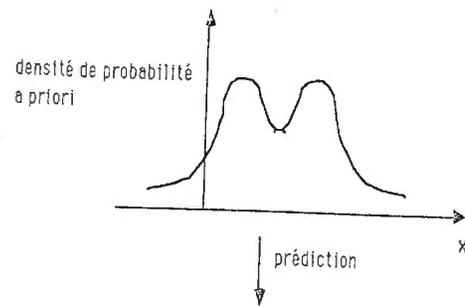
Comme l'état discret q_n est indépendant de $(x_k)_{k=0, \dots, n}$, nous voyons d'après (4.1) que la dp conditionnelle de transition et la dp conditionnelle d'observation ont la forme suivante:

$$TR_x(x_{k+1}, q_{k+1} | x_k, q_k) = tr_x(x_{k+1} | x_k, q_k) p_{t_k}(q_{k+1}, q_k) \quad (4.3)$$

$$tr_x(x_{k+1} | x_k, q_k) = N(x_{k+1}, A_k(q_k) x_k, G_k(q_k) Q G_k(q_k)^T)$$

$$OB_x(y_k, o_k | x_k, q_k) = ob_x(y_k | x_k, q_k) p_{o_k}(o_k, q_k)$$

$$ob_x(y_k | x_k, q_k) = N(y_k, C_k(q_k) x_k, H_k(q_k) R H_k(q_k)^T)$$



sur la dimension continue d'espace x , on observe deux modes qui se modifient du fait de la prédiction, puis de l'intégration de l'observation

Estimation récursive appliquée à une densité de probabilité multimodale

Maintenant si nous appliquons (3.6) avec (4.3):

$$p_{k+1|k}(x, q) = \sum_r \int p_{k|k}(t, x) \tau_{r_k}(x|t, x) dt p_{t_k}(q, r)$$

$$= \sum_{r, n} (p_{t_k}(q, r) \cdot A_n \cdot \partial(r, q_n) \int N(t, x_n, \Sigma x_n) N(x, A_k t, G_k Q G_k^t) dt)$$

En utilisant la formule de convolution pour les distributions gaussiennes, nous obtenons:

$$p_{k+1|k}(x, q) = \sum_{r, n} (p_{t_k}(q, r) \cdot A_n \cdot \partial(r, q_n) N(x, A_k x_n, A_k \Sigma x_n A_k^t + G_k Q G_k^t))$$

alors de (3.3) nous tirons:

$$p_{k+1|k+1}(x, q) = K_k \sum_{r, n} (p_{o_{k+1}}(o_{k+1}, q) \cdot p_t(q, r) \cdot A_n \cdot \partial(r, q_n) \cdot N(x, A_k x_n, A_k \Sigma x_n A_k^t + G_k Q G_k^t) \cdot N(y_{k+1}, C_{k+1} x, H_{k+1} R H_{k+1}^t))$$

où $A_k = A_k(r)$, $G_k = G_k(r)$, $C_{k+1} = C_{k+1}(q)$, $H_{k+1} = H_{k+1}(q)$. Mais nous remarquons que le produit de deux fonctions gaussiennes: $N(x, A_k x_n, A_k \Sigma x_n A_k^t + G_k Q G_k^t) \cdot N(y_{k+1}, C_{k+1} x, H_{k+1} R H_{k+1}^t)$ pourrait résulter du cas classique où la dp est une gaussienne normale, avec la normalisation en moins.

Le résultat est donc une gaussienne non normalisée, dont la moyenne et la matrice de covariance peuvent être calculées grâce aux formules d'itération de Kalman:

$$p_{k+1|k+1}(x, q) = K_k \sum_{r, n} B_{rnq} N(x, x_{rnq}, \Sigma_{rnq})$$

$$= \sum_{r, n} M_{rnq}(x)$$

où M_{rnq} , mode à l'instant $k+1$, fils du mode n de l'instant k : $(r, A_n, x_n, \Sigma x_n)$, est caractérisé par: $(q, K_k B_{rnq}, x_{rnq}, \Sigma_{rnq})$.

Bien sur, nous ne voulons pas que le nombre de modes croisse indéfiniment avec le temps,

donc, de la même manière que dans les systèmes de poursuite multicible, les hypothèses (modes) les moins probables (dont les poids sont les moins forts) doivent être éliminées de manière à en garder le nombre total constant. Une solution plus efficace en vue du contrôle est de fusionner ces modes faibles dans un mode à très grande dispersion, donc à très grande variance, de manière à quantifier la probabilité de l'hypothèse "je ne sais pas".

Et nous découvrons finalement les formules de mise à jour de ce genre de dp dans un temps discrétisé, en utilisant les formules de Kalman classiques. Il s'agit de calculer $(q, K_k, B_{rnq}, x_{rnq}, \Sigma_{rnq})$ à partir de $(r, A_n, x_n, \Sigma_{x_n})$, r et q étant les états de la paramétrisation discrète aux instants respectifs k et $k+1$.

A partir des formules classiques ((4.4), voir 4.1.4), nous savons déjà calculer les moyennes x_{rnq} et les matrices de covariance Σ_{rnq} des nouveaux modes:

$$x_{rnq} = A_k x_n + L_{k+1} [Y_{k+1} - C_{k+1}(q) A_k(q) x_n]$$

$$\Sigma_{rnq} = (I - L_{k+1} C_{k+1}(q)) \Sigma_{x_{k+1|k}}$$

avec:

$$\Sigma_{x_{k+1|k}} = (A_k(x) \Sigma_{x_n} A_k(x)^t + G_k(x) Q G_k(x)^t)$$

$$L_{k+1} = \Sigma_{x_{k+1|k}} C_{k+1}(q)^t [C_{k+1}(q) \Sigma_{x_{k+1|k}} C_{k+1}(q)^t + H_{k+1}(q) R H_{k+1}(q)^t]^{-1}$$

Pour calculer le poids $K_k \cdot B_{rnq}$ de chaque mode, nous pouvons écrire pour $x = 0$:

$$M_{rnq}(0) = K_k B_{rnq} N(0, x_{rnq}, \Sigma_{rnq})$$

soit

$$M_{rnq}(0) = K_k P^{o_{k+1}}(o_{k+1}, q) P^t_k(q, z) A_n \partial(x, q_n) \cdot N(0, A_k x_n, A_k \Sigma_{x_n} A_k^t + G_k Q G_k^t) \cdot N(y_{k+1}, 0, H_{k+1} R H_{k+1}^t)$$

donc

$$B_{rnq} = P^{o_{k+1}}(o_{k+1}, q) P^t_k(q, z) A_n \partial(x, q_n) \cdot N(0, A_k x_n, A_k \Sigma_{x_n} A_k^t + G_k Q G_k^t) \cdot N(y_{k+1}, 0, H_{k+1} R H_{k+1}^t) / N(0, x_{rnq}, \Sigma_{rnq})$$

Enfin, pour calculer le coefficient de chaque mode $M_{rnq}(x)$, il reste à normaliser les poids, en négligeant les modes les plus faibles surnuméraires:

$$K_k \cdot B_{rnq} = B_{rnq} / \left(\sum_{\substack{\text{sur les } p \\ \text{plus grands } B_{rnq}}} B_{rnq} \right)$$

L'algorithme est donc n fois plus complexe que le filtre de Kalman simple, où n est le nombre de modes, puisqu'il faut mettre à jour chaque mode, plus n fois le calcul de la constante de normalisation.

4.1.4 Annexe: formules d'itération de Kalman classiques

le modèle markovien linéaire-gaussien classique est:

$$x_{k+1} = A_k x_k + G_k w_k$$

$$y_k = C_k x_k + H_k v_k$$

x_k, y_k étant respectivement les vecteurs d'états et d'observation dans des espaces réels multidimensionnels, v_k et w_k étant des variables aléatoires gaussiennes non corrélées de dp respectives $N(0, R)$ et $N(0, Q)$. A_k, G_k, C_k et H_k sont des matrices réelles de dimensions appropriées. La mise à jour de la dp gaussienne $N(x_k, \Sigma_{x_k})$ grâce à (3.7) donne les formules d'itération suivantes:

$$x_{k+1} = A_k x_k + L_{k+1} [y_{k+1} - C_{k+1} A_k x_k] \quad (4.4)$$

$$\Sigma_{x_{k+1}} = (I - L_{k+1} C_{k+1}) \Sigma_{x_{k+1|k}}$$

$$\Sigma_{x_{k+1|k}} = (A_k \Sigma_{x_k} A_k^t + G_k Q G_k^t)$$

$$L_{k+1} = \Sigma_{x_{k+1|k}} C_{k+1}^t [C_{k+1} \Sigma_{x_{k+1|k}} C_{k+1}^t + H_{k+1} R H_{k+1}^t]^{-1}$$

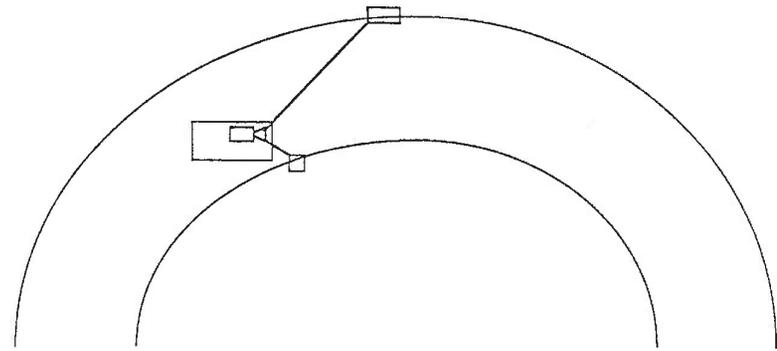
Ces équations récursives réalisent le calcul fonctionnel (3.7), sachant que si la k ème dp est une gaussienne, exactement décrite par $(x_k, \Sigma x_k)$, alors la $k+1$ ème sera aussi une gaussienne, décrite par $(x_{k+1}, \Sigma x_{k+1})$ [Kumar & Varaiya].

4.1.5 Application et conclusion sur cet algorithme

Imaginons que pour la conduite d'un robot mobile sur route, comme dans l'exemple du début en 1.1.1, on utilise deux petites fenêtres de traitement temps réel dans l'image fournie par la caméra sur le véhicule, fenêtres destinées à suivre les bords de la route. De plus, on utilise pour asservir ces fenêtres en poursuite un filtrage de Kalman, en approximant à un instant donné l'évolution de la voiture et la projection des limites de la route dans l'image par un modèle linéaire. Ce genre de technique est réellement utilisé sur des véhicules autonomes [Dickmanns & Zapp]. Imaginons encore que la seule perception disponible soit à travers cette image, ce qui est cette fois plus irréaliste (mais l'algorithme est surtout intéressant pour les systèmes mal observés).

Bien sûr, on peut déjà imaginer utiliser cet algorithme pour modéliser les défauts du traitement d'image, par exemple la non-détection du bord de la route, soit pour des raisons spécifiques au traitement (bruit, erreur), soit à cause d'effets de masquage, d'ombrage, à cause d'une tache... Mais cela n'apporte guère d'amélioration, puisqu'en l'absence d'observation tout bon filtre de Kalman boucle simplement sur la prédiction, sans avoir à incorporer un modèle complexe de transitions discrètes. On peut à la rigueur s'en servir dans ce cas pour la détection de l'absence du bord de la route (masquage, arrivée à un embranchement).

Une autre application plus intéressante est l'incorporation de la possibilité de glissement dans un virage. Si par exemple la route est mouillée ou glissante, ce qui pourrait constituer la partie discrète de la perception (détection de pluie, de flaques, de plaques de verglas...), il y a danger de dérapage dans les virages, d'autant plus grand que l'effort latéral est grand sur les roues arrière (dans le cas d'un survirage). On se trouve alors bien devant deux modèles d'évolution différents, séparés par une transition discrète, et imposant des comportements de contrôle différents (contre-braquage en cas de glissement). Le filtre de Kalman doit en tenir compte sous peine d'erreurs graves dans la conduite en cas d'une fausse estimation.



contrôle de véhicule fonction d'un suivi des bords de route et d'un modèle mixte avec/sans glissement

D'autre part, dans le cas d'un pilotage fin autour de la limite de glissement (robot pilote de course ?!), le filtre permettrait de traiter simultanément les deux hypothèses du modèle, avec éventuellement une prise de décision de contrôle (volant, etc..) pondéré par leurs probabilités dans les cas douteux.

Cet algorithme, qui n'apporte rien de nouveau dans la technique de traitement du signal, vise donc à montrer la possibilité de nouveaux traitements probabilistes décrivant les paramètres de l'univers (de plus en plus complexes lorsque les robots évoluent dans des environnements moins contrôlés) par des distributions de probabilités non classiques, incorporant notamment des aspects discrets, et des possibilités de changement de modèle. D'autre part, en intégrant ces notions proches du haut niveau, qui s'apparentent à des descriptions qualitatives de la situation (glissement ou non), et des observations qualitatives (route mouillée), on se rapproche d'une utilisation robotique, avec interfaçage plus aisé à un système intelligent, et intégration d'informations a priori ou externes (par exemple la météo).

4.2 Contrôle d'une fenêtre d'observation

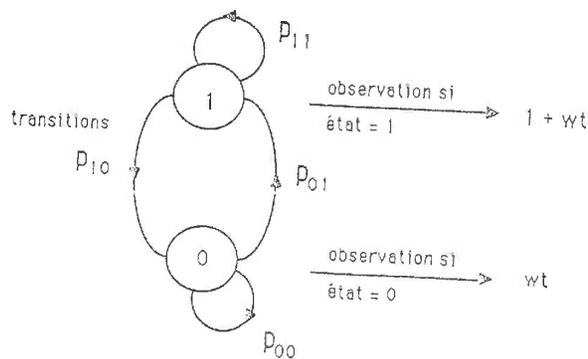
Le premier scénario de contrôle de la perception que nous allons étudier est un schéma extrêmement simplifié de surveillance d'un horizon par un radar ou par une caméra. Il ne possède en effet qu'un capteur, et l'univers monodimensionnel observé est composé d'une suite finie de 0 et de 1. Mais nous allons voir qu'il présente une certaine complexité, et qu'il est déjà intraitable par le théorème des "bandit processes", dont nous utiliserons cependant les résultats d'une manière approchée.

Dans ce modèle, nous illustrerons tout d'abord l'intérêt des stratégies bayésiennes de perception, et notamment de l'optimisation de la commande du capteur. Puis nous étudierons le comportement des algorithmes utilisés, notamment en fonction des approximations introduites dans les probabilités a priori, et dans les calculs d'intégration et d'optimisation. Ensuite, nous montrerons comment l'étendre à un scénario beaucoup plus réaliste, au prix d'une certaine complexification, en nous rapprochant en particulier d'un schéma d'optimisation du balayage électronique pour un sonar ou un radar de surveillance, en utilisant toujours les mêmes techniques d'optimisation du contrôle du faisceau.

Nous développerons enfin au chapitre suivant un dernier modèle, utilisant plusieurs capteurs, avec un environnement plus complexe, destiné à ébaucher le schéma d'un système plus complet.

4.2.1 Le modèle

Le problème posé est le contrôle de déplacement d'une fenêtre d'observation sur une scène 1D afin d'optimiser la surveillance, au sens de la maximisation de la quantité d'information contenue dans le résultat de l'estimation réursive de la scène.



modèle d'évolution et d'observation d'un pixel
Xavier Merlo Mai 18 1988

La scène est une suite de n pixels $(x^i)_{i=1,..,n}$, chacun pouvant prendre les valeurs 0 ou 1, correspondant respectivement à l'absence ou la présence d'un objet sur l'horizon surveillé, et évoluant dans le temps discrétisé de manière indépendante suivant un automate probabiliste. De cette manière la scène évolue dynamiquement, les objets apparaissant et disparaissant de manière poissonnienne, mais ne pouvant dans ce modèle se déplacer. La scène a de plus été rebouclée sur elle-même, le pixel suivant x^n étant le pixel x^1 , ce qui explique les modulus dans les formules suivantes. Nous examinerons à la fin de ce chapitre les problèmes, notamment de combinatoire des calculs, posés par une modélisation plus complexe, où par exemple les objets peuvent se déplacer sur l'horizon, est groupé en cortège, ou être de grande taille. Les pixels ne sont alors plus indépendants les uns des autres, et nous serons naturellement amenés à introduire la notion de hiérarchisation en niveaux exposée au chapitre II, importante en intelligence artificielle.

Le modèle du robot de surveillance n'a pas accès directement au vecteur d'état $(x_t^i)_{i=1,..,n}$ à l'instant t , mais à travers un vecteur d'observation $(y_t^i)_{i=1,..,n}$. Comme la scène est observée à travers une fenêtre de lf pixels à partir du pixel f , tous les y^i tels que $i < f$ ou $i > f + lf$ sont nuls, les autres renvoyant une variable aléatoire y de densité de probabilité $p(y|x)$. Pour fixer les idées dans le modèle, et sans référence à un cas réel, on a pris une distribution gaussienne autour de x (bruit additif gaussien w_t^i):

$$y_t^i = x_t^i + w_t^i \quad \text{si } f \leq i < f+lf \text{ ou } (f+lf) < n \text{ et } i < f+lf \text{ mod } n$$

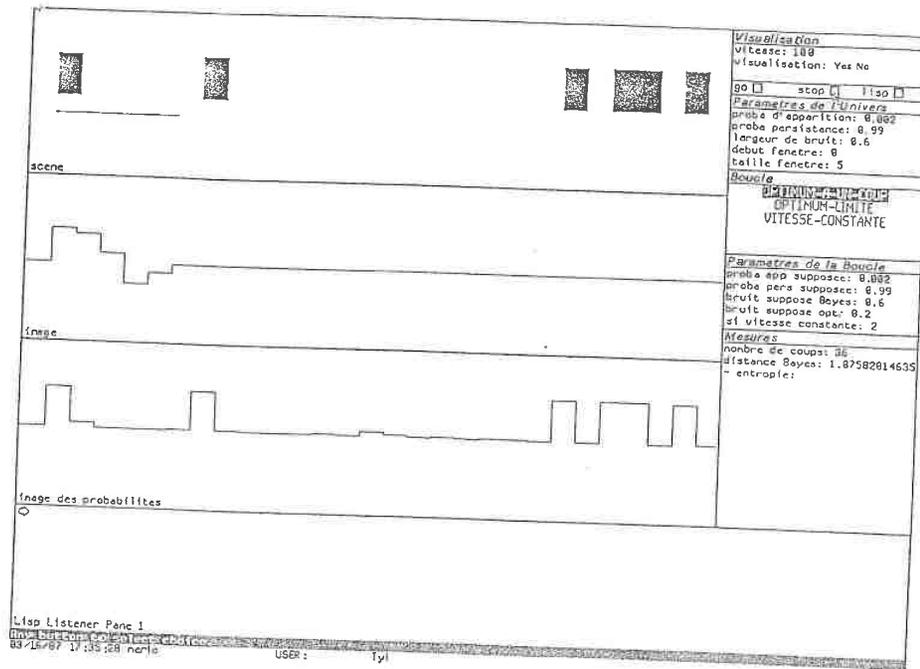
$$= 0 \quad \text{sinon}$$

toutes les w étant des variables aléatoires gaussiennes de variance v indépendantes entre elles en temps et en espace.

La position f de la fenêtre d'observation est déterminée à chaque instant par une stratégie de contrôle, de manière à balayer régulièrement tout l'horizon et à minimiser les erreurs de détection, comme cela sera décrit plus précisément plus loin.

4.2.2 Estimation réursive du vecteur d'état

Les observations sont filtrées à chaque instant par une estimation réursive, du type de celle détaillée au chapitre III. Les informations sont intégrées dans la densité de probabilité des états grâce à la formule de Bayes appliquée à la perception. Dans ce modèle, on ne cherche pas directement à optimiser l'estimation, qui, dans l'estimation réursive, n'intervient pas dans la boucle prédiction-intégration des observations, mais en sortie de celle-ci. Il s'agit plutôt d'optimiser en moyenne la quantité d'information intégrée. L'écart moyen de l'estimateur du maximum de vraisemblance peut d'ailleurs être un des critères de cette mesure, parmi d'autres que l'on examinera.



visualisation de la simulation

La fenêtre de visualisation de la simulation est séparée en plusieurs parties: celle marquée "scène" correspond à la scène $(x^i)_{i=1, \dots, n}$ du modèle, celle marquée "image" correspond à l'observation $(y^i)_{i=1, \dots, n}$ qui en est faite, et celle marquée "image des probabilités" correspond au tableau $(p^i)_{i=1, \dots, n}$ des probabilités élémentaires. Sur la droite, se trouvent visualisés différents paramètres de contrôle du modèle, le choix de l'algorithme de contrôle, le choix de ses paramètres (dont les hypothèses sur les probabilités a priori de la scène, qui ne sont pas forcément exactes comme on le verra dans la suite), et enfin des mesures de performance sur la boucle de contrôle courante. Comme tous les pixels évoluent et sont observés de manière indépendante les uns des autres, la densité de probabilité sur les états de $\{0, 1\}^n$ se ramène au produit des probabilités sur chaque pixel:

$$P(x^1=u^1, x^2=u^2, \dots, x^n=u^n) = \prod_{i=1}^n P(x^i=u^i)$$

On réduit donc l'espace des densités de probabilité à $\{0, 1\}^n$, ce qui permet de réduire la taille du problème de 2^{n-1} à n . On notera $(p_t^i)_{i=1, \dots, n}$ le vecteur de ces probabilités élémentaires, avec $p_t^i = P(x_t^i=1)$. C'est cette approximation d'indépendance a priori des pixels qui rend tous les calculs simples. La prédiction donne alors pour chaque pixel i , en notant $(p_t^i)_{i=1, \dots, n}$ le vecteur des probabilités à l'instant t , prédites à partir des observations $(y_k^i)_{k=0, \dots, t-1}$:

$$p_t^i = P(x_t^i=1 | y_0^i, \dots, y_{t-1}^i)$$

$$p_t^i = p_{t-1}^i P_{11} + (1-p_{t-1}^i) P_{01}$$

Ces probabilités prédites à l'instant t à partir des observations à $t-1$ vont servir de probabilités a priori dans l'estimation récursive. Soit pour les probabilités a posteriori (après observation) $(p_t^i)_{i=1, \dots, n}$:

$$p_t^i = p_{t-1}^i \cdot p(y_t^i | x_t^i=1) / \text{norm} \quad \text{si } f_{t+1} \leq i < f_{t+1} + l_f$$

$$= p_{t-1}^i \quad \text{sinon}$$

et

$$\text{norm} = p_{t-1}^i p(y_t^i | x_t^i=1) + (1-p_{t-1}^i) p(y_t^i | x_t^i=0)$$

4.2.3 Optimisation du contrôle du déplacement de la fenêtre

On cherche à maximiser en moyenne la quantité d'information contenue dans la densité de probabilité. Il s'agit en fait de considérer le système d'estimation récursive comme un tout à optimiser. Le nouvel espace des états est alors celui de l'ensemble des probabilités p^i , soit $\{0, 1\}^n$, l'évolution étant donnée par les opérations de prédiction et d'intégration du filtrage. On passe donc d'un système initial imparfaitement observé à un système parfaitement observé dans un espace plus complexe (l'espace des informations ou des "information states" [Kumar & Varaiya]). Comme en effet à chaque instant toute l'information acquise sur le système initial est condensée dans la densité de probabilité, un contrôle optimal sera seulement fonction de celle-ci:

$$g_{opt}: \quad \begin{array}{l} \{0, 1\}^n \rightarrow \{1, 2, \dots, n\} \\ (p^i) \rightarrow f = g_{opt}(p^i) \end{array}$$

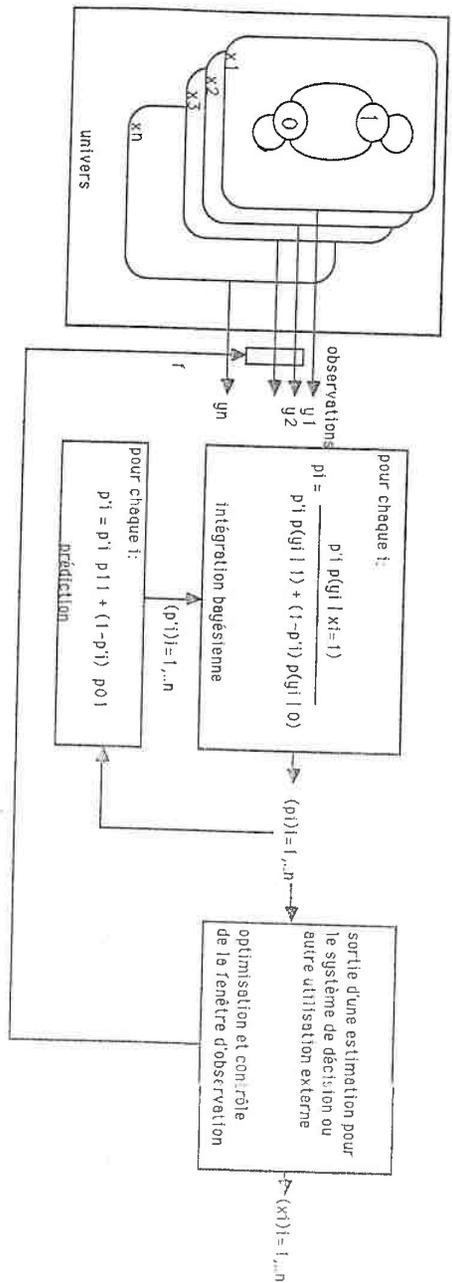


schéma de l'estimation récursive du modèle

A chaque instant k, pour chaque pixel, les équations d'évolution de ce nouveau système (ou méta-système) sont s'il n'y a pas observation:

$$P_{k+1} = \mathbb{E}(P_k) = P_k P_{11} + (1-P_k) P_{01} = P_k (P_{11}-P_{01}) + P_{01} \quad (4.5)$$

et s'il y a observation:

$$P_{k+1} = O(p_k, y) = p_k P(y|1) / (p_k P(y|1) + (1-p_k) P(y|0)) \quad (4.6)$$

où $p_k = P_k (P_{11}-P_{01}) + P_{01}$, y est la valeur prise par une variable aléatoire Y_{k+1} réelle de densité de probabilité: $(p_k P(y|1)+(1-p_k) P(y|0))$, et $P(y|1)$ et $P(y|0)$ sont des gaussiennes en y de variance v_b , de moyennes respectives 0 et 1.

On peut alors se fixer différents critères de performance ou d'erreur, décrivant le caractère plus ou moins "pointu" des probabilités d'existence d'objet en chaque point. On peut utiliser les différents coûts élémentaires suivants:

Ecart du maximum de vraisemblance:

$$C(p) = \mathbb{E} \left(\sum_i (x^i=0 \text{ et } p^i > 1/2) + (x^i=1 \text{ et } p^i < 1/2) \right) \\ = \sum_i \min(1-p^i, p^i)$$

Ecart des probabilités:

$$C1(p) = \mathbb{E} \left(\sum_i p^i (1-x^i) + (1-p^i) x^i \right) \\ = \sum_i 2 p^i (1-p^i)$$

Comme nous cherchons à minimiser la moyenne du coût dans le temps, le vrai critère à optimiser doit prendre en compte ces coûts élémentaires d'une manière analogue à celle exposée au chapitre 3, sur plusieurs instants consécutifs, ou même à la limite sur un horizon infini. La pondération accordée à chaque instant permet le réglage de la tâche à optimiser: un poids relatif important sur les instants immédiats, donc en particulier dans J2, si β est faible, permet de privilégier la rapidité de réaction de l'estimation aux dépens de l'erreur moyenne. En effet, les poids plus faibles sur les instants éloignés minimisent leur importance dans le critère de coût.

$$J1(u) = \lim_{N \rightarrow \infty} 1/N \mathbb{E} \left(\sum_{n=0}^N C(p_n) \right)$$

$$J2(u) = E \left(\sum_{n=0}^{\infty} \beta^n C(P_n) \right) \quad \beta < 1$$

$$J3(u) = E \left(\sum_{n=0}^N C(P_n) \right)$$

$u = (u_k)_{k=1, \dots, \infty}$ étant un contrôle donné de f , que l'on cherche à optimiser. L'espérance est prise sur les réalisations des x_0^i , des w_t^i , et des transitions dans les automates. Les p_i sont bien sur mis à jour à l'aide des formules (3.3) et (3.4), qui ont donné (4.5) et (4.6).

On est alors ramené à un problème de contrôle pur, la variable de commande étant la position de la fenêtre d'observation, et l'état du système étant l'ensemble des probabilités d'existence d'objet dans chaque case. Malheureusement, les méthodes pour traiter de telles fonctions de coût sur un horizon infini opèrent plutôt sur des espaces à nombre d'états fini, de même que la programmation dynamique, qui est classique pour un horizon fini. Une discrétisation raisonnable de $[0,1]^n$ paraissant ici impossible, de telles méthodes ne peuvent être utilisées.

On peut être tenté de se rapprocher du modèle de "bandit processes" (voir 3.6.1), ce qui nous permettrait de prouver l'existence d'une méthode simple de décision optimale, et éventuellement de la construire. En effet, assimilons chaque case ou pixel du modèle de surveillance à un des processus stochastiques élémentaires décrits au chapitre III. Si on limite la largeur de la fenêtre I_f à 1, la ressource à partager entre tous les processus élémentaires est bien sur l'observation à travers cette fenêtre, la commande c correspondant à l'allocation de celle-ci.

Un problème qui se pose est qu'un processus élémentaire évolue même si n'y a pas observation, puisque la prédiction modifie la valeur de la probabilité courante d'existence d'un objet, ce qui fait que l'on sort des hypothèses du théorème. Etudions donc d'abord le problème de l'observation si la scène ne change pas, c'est à dire si $p_{11}=1$ et $p_{01}=0$. On se trouve alors dans les hypothèses du théorème DAI, chaque pixel étant identifié à un processus stochastique (Ω, U, P, R, β) , avec $\Omega=[0,1]$ espace des états, $U=\{\text{"observé"}, \text{"caché"}\}$ ensemble des commandes, P la loi probabiliste d'évolution, β est le facteur de décroissance exponentiel.

R est la fonction récompense à maximiser: ici $R(p) = C(p) - C(O(p,y))$, $O(p,y)$ étant la probabilité a posteriori tenant compte de l'observation y . Montrons que minimiser la fonction de coût $J2(u)$ est équivalent à maximiser $J4(u) = E(\sum \beta^k R(p_{uk}))$ (en effet dans le modèle du bandit à plusieurs bras, à un instant seul le processus actionné donne lieu à une récompense, alors que dans $J2$, tous les pixels contribuent):

$$J2(u) = \sum_{k=0}^{\infty} \beta^k \sum_{i=1}^n C(P_k^i)$$

$$\sum_{i=1}^n C(P_k^i) = \sum_{i=1}^n C(P_k^{i-1}) - R(P_k^{uk})$$

$$= \sum_{i=1}^n C(P_0^i) - \sum_{j=1}^k R(P_j^{uj})$$

Donc minimiser $J2(u)$ est équivalent à maximiser:

$$\begin{aligned} \sum_{k=0}^{\infty} \sum_{j=0}^k \beta^k R(P_j^{uj}) &= \sum_{j=0}^{\infty} R(P_j^{uj}) \sum_{k=j}^{\infty} \beta^k \\ &= \sum_{j=0}^{\infty} R(P_j^{uj}) \beta^j / (1-\beta) \end{aligned}$$

Ce qui prouve qu'il est bien équivalent de maximiser $J4(u)$ et de minimiser $J2(u)$.

Pour le système constitué de l'ensemble $((\Omega_i, U_i, P_i, R, \beta))_{i=1, \dots, n}$ des pixels, le problème est donc de choisir à chaque instant la position f de la fenêtre d'observation. On aura alors les commandes élémentaires $u_i = \text{"observé"}$ et $u_i = \text{"caché"}$ pour $i \neq f$. Le théorème nous dit donc qu'il existe une fonction $\mu: [0,1] \rightarrow R$, telle que pour un état $(p^i)_{i=1, \dots, n}$ du système, la commande:

$$f = \arg \max_{i=1, \dots, n} \mu(p^i)$$

est optimale au sens de la maximisation de:

$$E \left(\sum_{k=0}^{\infty} \delta^k R(p_k, f) \right)$$

De plus on sait que:

$$\mu(p) = \sup_{t > 0} \frac{E \left(\sum_{k=0}^{t-1} \delta^k R(p(k)) \right)}{E \left(\sum_{k=0}^{t-1} \delta^k \right)}$$

avec $p(0) = p$ et $p(k) = O(p(k-1), y_k)$.

Si maintenant on étudie le cas où $p1-1 \ll 1$ et $p01 \ll 1$, le temps caractéristique d'évolution de la scène est relativement long devant celui de l'observation, ce qui correspond en général assez bien avec des modèles réels de tâche de surveillance. On a donc une certaine continuité dans l'évaluation de la stratégie. L'indice d'allocation dynamique $\mu(p)$ donne donc une stratégie "assez bonne" ou "quasi optimale".

L'extension à une largeur de la fenêtre supérieure à 1 est plus délicate, puisqu'on sort des hypothèses de base du théorème et qu'on n'a plus d'indication de continuité. Néanmoins, si on restreint la commande de position de la fenêtre à des valeurs multiples de lf , ce qui revient à grouper les pixels par paquets de taille lf , alors l'indice:

$$\mu_f(i) = \sum_{j=i}^{i+lf-1} \mu(p_j)$$

permet de déterminer une stratégie optimale sous la restriction vue plus haut. La stratégie déterminée par $\mu_f(i)$ sans restriction de position sur la fenêtre, est donc là aussi intuitivement une "assez bonne" stratégie. En fait, dans une première approximation, elle correspond à une optimisation de l'espérance du coût sur le coup suivant, donc sur un horizon de 1. Nous avons vu qu'on ne pouvait pas utiliser la programmation dynamique pour minimiser le coût sur deux coups ou plus. Par contre, il est possible de réaliser cette minimisation sur un horizon fini par développement de l'arbre des possibilités de commande. Le coût en calcul devient cependant grand dès que l'on dépasse un horizon de deux ou de trois coups.

La complexité de l'algorithme tel qu'il est utilisé dans ce modèle est faible, puisqu'elle est de n , où n est la taille de la discrétisation de l'horizon, grâce à la notion d'indice d'allocation dynamique.

4.2.4 Les résultats

La justification "physique" ou approchée du modèle de contrôle proposé ci-dessus n'est bien sûr pas suffisante, et puisque la démarche de départ exposée dans l'introduction se veut pragmatique et non théorique, une validation sur une simulation était nécessaire, à défaut d'application réelle de ce type de contrôle perceptif au moment de la rédaction de ce mémoire.

Les performances du modèle proposé sur une simulation simple ont été évaluées grâce des critères numériques, par comparaison avec des stratégies de contrôle plus classiques. Dans le problème du contrôle de la fenêtre d'observation, ces techniques classiques de référence sont des balayages réguliers de la scène par la fenêtre, d'une manière analogue au balayage circulaire d'un sonar ou d'un radar de veille.

Nous pouvons donc quantifier, dans notre modèle simple de simulation les gains apportés par l'optimisation du contrôle perceptif. D'autre part, plusieurs algorithmes ont été testés, avec différents types de contrôle possibles (un contrôle avec "inertie" de la fenêtre et limitation de son déplacement, et un sans limitation analogue au contrôle du faisceau d'un balayage électronique), ce qui permet d'effectuer d'intéressantes comparaisons. Enfin, nous avons évalué la baisse de performance due aux erreurs sur les probabilités a priori de transition et d'observation du modèle de l'univers, de manière à valider également le choix des techniques bayésiennes.

Deux erreurs élémentaires moyennes ont été mesurées: écart quadratique de la probabilité à la réalité $Q(p)$, écart du maximum de vraisemblance à la réalité $MV(p)$, en faisant tourner des simulations suffisamment longues (nombre d'itérations supérieur à 100.000). L'erreur moyenne de l'estimateur du maximum de vraisemblance est le vrai critère d'erreur d'un système de surveillance, qui est capable de fournir à chaque instant une estimation de la situation. L'erreur quadratique mesure plus les probabilités elles-mêmes, dans leur écart avec la connaissance parfaite ($p=1$ s'il y a un objet, $=0$ sinon). Un tel critère d'erreur peut être une indication utile si la finalité du système n'est pas la surveillance (comme dans le modèle exposé au chapitre suivant, s'il s'agit d'une tâche de détection au plus tôt d'un objet), puisqu'il estime la qualité des probabilités mises à jour dans l'estimation récurrente.

Les courbes ci-dessous donnent les résultats de ces simulations, effectuées avec les paramètres de l'univers suivants:

probabilité de persistance $p11 = 0,99$
 probabilité d'apparition $p01 = 0,002$

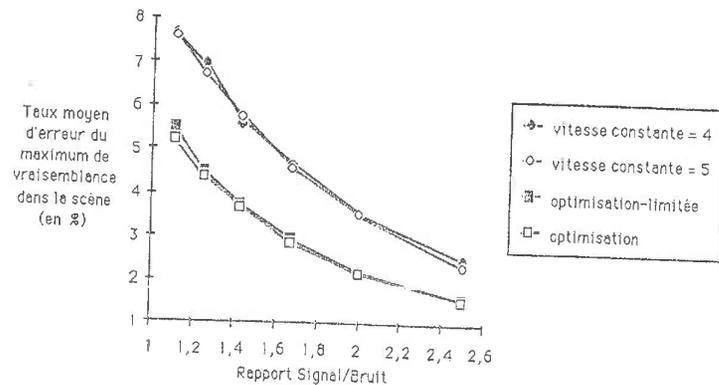
La taille de la discrétisation de l'horizon est de 29 (nombre de pixels). Le taux de couverture de l'observation est donc de $5/29 = 17\%$ (rapport de la taille de la fenêtre d'observation à la taille de l'horizon).

Dans ces premières courbes, on compare les performances respectives des deux types de stratégie: balayage régulier, avec les vitesses donnant les meilleurs résultats, ou balayages optimisés de différents types:

- "Optimisation" correspond à la stratégie optimisée vue plus haut. En fonction de la valeur en chaque pixel de l'indice, on choisit la position de la fenêtre au coup suivant qui maximise la somme des indices des pixels observés.

- "Optimisation-limitée" correspond à la même stratégie. Simplement, afin de simuler une "inertie" de la fenêtre, on restreint les positions autorisées à l'instant suivant dans une certaine marge autour de la position courante.

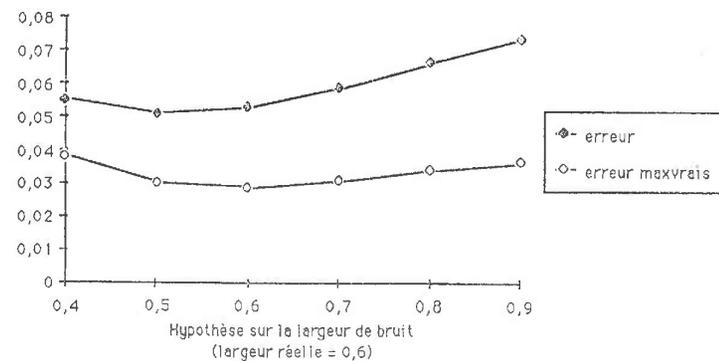
Les paramètres pris en compte dans l'estimation récurrente, sont exactement les mêmes que les paramètres de la simulation. Les hypothèses a priori sur le modèle de l'univers dans cette première série d'essais sont donc exactes.



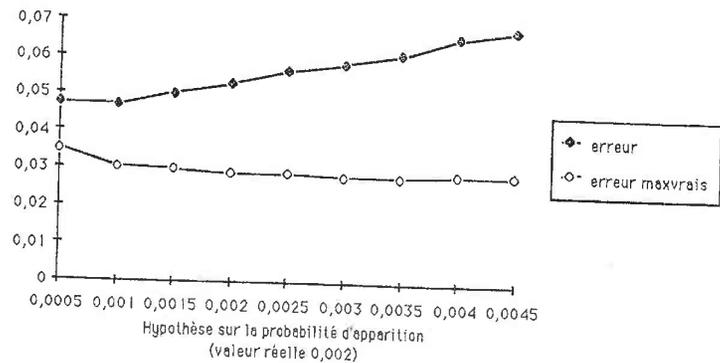
Performances des différentes stratégies en fonction du rapport signal/bruit

On remarque donc que les stratégies optimisées sont meilleures que les stratégies à balayage régulier, et ce quelque soit la variance du bruit gaussien. Pour comparer mieux les performances respectives des deux types de stratégie, on peut arriver à une indication de l'amélioration de la performance en terme du gain sur le rapport signal/bruit qui conduit à la même performance. Ainsi, les stratégies d'optimisation arrivent avec un bruit de largeur 0,7 (rapport $S/B = 1,4$) à la même performance que les stratégies de balayage régulier avec un bruit de largeur 0,5 (rapport $S/B = 2$ pour un même signal de niveau 1), soit un gain en rapport signal/bruit significatif, bien sur, dans le modèle simpliste utilisé. Il s'agit néanmoins d'une indication intéressante, d'autant plus que l'on constate que ce gain s'accroît au fur et à mesure que le niveau du bruit devient plus important.

Les mesures de performance suivantes de la méthode d'optimisation du contrôle ont été faites avec les mêmes paramètres de l'univers que précédemment (le niveau du bruit étant fixé à 0,6), mais avec des hypothèses sur cet univers volontairement erronées pour l'estimation récurrente. On peut ainsi évaluer les effets des erreurs sur les probabilités a priori pour un critère de performance global. Le premier tableau illustre la variation de l'erreur moyenne d'une stratégie optimisée en fonction de l'hypothèse sur le niveau de bruit, et le deuxième tableau en fonction de l'hypothèse sur la probabilité d'apparition $p01$. Dans un scénario plus réaliste, en effet, c'est cette probabilité d'apparition qui est la plus incertaine, la probabilité de disparition étant reliée directement à la durée moyenne de la présence des objets dans l'environnement.



Erreurs moyennes quadratique et du maximum de vraisemblance en fonction de l'hypothèse sur le niveau de bruit



Erreurs moyennes quadratique et du maximum de vraisemblance en fonction de l'hypothèse sur la probabilité d'apparition des objets

L'erreur moyenne de l'estimateur du maximum de vraisemblance reste remarquablement peu sensible aux erreurs des hypothèses a priori. On a ainsi une indication de la validité des techniques bayésiennes pour une tâche de surveillance. Par contre, la qualité moyenne des probabilités courantes se dégrade plus nettement, comme le montre la dégradation plus nette et plus rapide de l'erreur sur les probabilités. On remarque également que les minima d'erreur n'ont pas forcément lieu lorsque l'hypothèse est égale à la valeur réelle. Dans la mesure où l'optimisation du contrôle réalisée n'est qu'une approximation, une légère dérive est parfaitement compréhensible, d'autant plus que le minimum correspond, comme on peut le voir, à une zone particulièrement plate de la courbe d'erreur.

4.2.5 Commentaires, extensions, applications

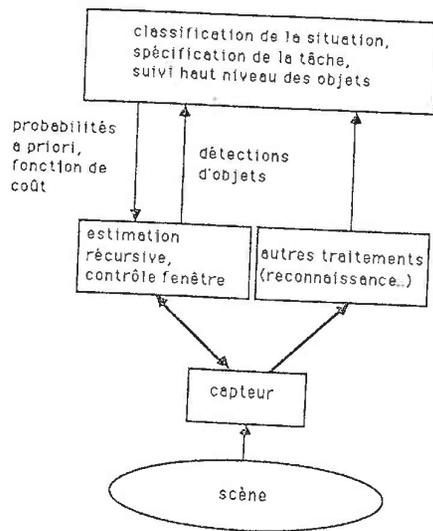
A travers ce petit modèle très simple, nous pensons avoir montré l'intérêt des techniques d'optimisation du contrôle perceptif, puisque nous arrivons à améliorer la performance de stratégies classiques de balayage plus systématique. Cette amélioration se traduirait bien sur pour un système radar ou sonar réel par un gain en rapport signal/bruit, à condition d'arriver à adapter ce type de méthode dans un modèle réaliste beaucoup plus complexe. Ce problème de fenêtre d'observation est en effet assez courant dans le domaine du traitement du signal, qu'il s'agisse d'une fenêtre spatiale, temporelle, ou fréquentielle. Par exemple dans un problème de surveillance de l'espace hertzien, la ressource à allouer peut en effet aussi bien être un filtre dont on peut faire à la fois varier la fréquence de base et la largeur. D'autre part, on remarque que l'erreur moyenne du maximum de vraisemblance varie beaucoup plus lentement que la qualité des probabilités courantes de l'estimation récursive, ce qui plaide en faveur d'une utilisation étendue des techniques bayésiennes.

Une première extension du scénario décrit plus haut permettrait déjà de se rapprocher beaucoup plus d'un modèle réaliste. Il suffirait d'utiliser des sources multiniveaux dans le modèle de l'univers, de manière à reproduire les effets conjugués du niveau d'émission et de l'atténuation due à la distance sur le niveau du signal réel. Cette extension peut être réalisée assez grossièrement si on se contente d'un petit nombre de niveaux, par exemple 4. Dans ce cas la probabilité élémentaire p^i serait remplacée par un vecteur de 4 probabilités. Hormis cette multiplication de la dimension de l'espace des probabilités, tous les calculs faits au 4.2.3 seraient toujours applicables, et la même optimisation pourrait alors être réalisée. La complexité des calculs augmenterait alors d'un facteur $4^2=16$, puisqu'il faudrait alors envisager toutes les transitions dans le nouvel automate à 4 états. Un compromis supplémentaire serait alors effectué par l'optimisation, celui de passer plus de temps sur les cibles faibles que sur celles à fort signal, sauf intérêt particulier accordé à ces dernières dans la fonction de coût (cible proche = plus importantes).

Une autre amélioration qui permettrait de rendre le modèle de surveillance plus réaliste, mais qui le remettrait en cause plus fondamentalement, serait de prendre en compte les possibilités de déplacement des objets. Ainsi un objet en j pourrait passer en $j-1$, ce qui contredirait alors l'hypothèse simplificatrice faite dès le départ, de l'indépendance de tous les pixels. Le nombre d'états envisageables passerait alors de $2n$ à 2^n , suivant un accroissement combinatoire typique. Une première approximation serait alors de traiter des voisinages de point, puisque l'influence des objets est locale si les vitesses sont faibles. Mais il existe de toute façon une autre manière de traiter ces problèmes de combinatoire: ce modèle présente un défaut manifeste si l'on prend en compte ce qui a été dit au chapitre II. Il travaille en effet sur un espace des états à plat, sans hiérarchiser la perception en différents niveaux, et sans utiliser la notion d'objet. Or c'est indispensable dès que l'on veut représenter toute la complexité de l'environnement, par exemple, comme on l'a vu au chapitre III, dès que l'on veut intégrer des attributs (vitesse, caractéristiques..) à une détection d'objet. Ce modèle n'est donc valable que localement, appliqué à une boucle de contrôle de la perception particulière. Nous allons essayer dans le chapitre suivant de développer un scénario de simulation plus complet intégrant mieux tous ces aspects.

4.2.6 Liaison avec un système d'intelligence artificielle

Non seulement des aspects et des représentations du type intelligence artificielle peuvent être intégrés dans un modèle probabiliste, comme on vient de le voir ci-dessus ou comme le filtre de Kalman modifié exposé en 4.1, mais le contrôle de la fenêtre d'observation capteur est également particulièrement adapté au fonctionnement en liaison avec un système travaillant à partir de données de haut niveau, traitant des informations qualitatives sur la situation ou sur le contexte. Une autre tâche pouvant être prise en compte par ce haut niveau est l'analyse qualitative du comportement des objets (apparitions et disparitions, déplacements), dont il peut déduire une classification de la situation.



superposition d'un bas niveau probabiliste et d'un niveau logique

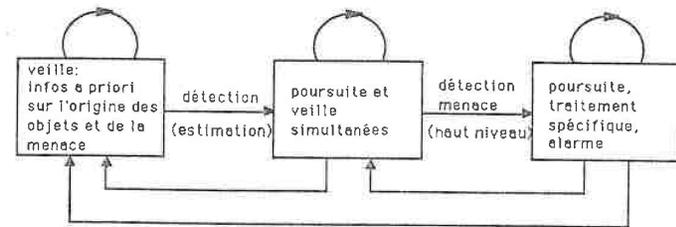
Deux types d'informations sont en effet nécessaires pour le paramétrage du module de contrôle de la fenêtre: les probabilités a priori d'existence, d'apparition et de disparition, et la fonction de coût suivant laquelle est faite l'optimisation.

Les probabilités a priori peuvent être issues:

- soit d'informations externes au système (zone la plus probable d'apparition des objets, données issues d'autres capteurs ou d'autres sources)
- soit d'une analyse haut niveau de l'historique du comportement des objets détectés: par exemple, les objets peuvent avoir tendance à se déplacer par groupe, ce qui fait qu'une détection renforce les probabilités a priori d'apparition dans son voisinage, ou un objet disparu récemment est susceptible de réapparaître.

Ces tables de probabilités a priori dépendent donc à la fois des informations initiales et de l'historique des objets détectés.

La fonction de coût est déterminée par la classification haut niveau de la situation, c'est à dire par le contexte: suivant le cas, on préférera faire de la veille simple, de la veille et de la poursuite simultanées (certains objets pouvant présenter un intérêt plus grand que d'autres, et un mélange pondéré entre la veille et la poursuite pouvant être réalisé), de la poursuite pure (seul compte alors le suivi de certains objets, au détriment de nouvelles détections), ou la prise en compte d'autres types de traitement à travers la fenêtre d'observation capteur, comme par exemple des traitements de reconnaissance des formes.



classification qualitative en trois contextes

Cette classification haut niveau correspond tout d'abord à des situations qualitativement différentes: veille, présence d'un ou plusieurs objets, présence d'une menace reconnue. Mais le choix d'une tâche perceptive, donc de fonction de coût peut également prendre en compte un caractère probabiliste des hypothèses ainsi qu'une notion de risque. Prenons l'exemple d'un système de détection de menace aérienne (missile, avion, hélicoptère...), utilisant notre modèle de contrôle d'une fenêtre d'observation comme un bas niveau probabiliste. Face aux diverses menaces, graduées qualitativement (type) et quantitativement (imminence et incertitude de l'attaque), il dispose bien sûr d'un ensemble de ripostes possibles, caractérisées par une efficacité et un délai d'intervention, donnés pour un type d'attaque.

On se retrouve dans une situation de jeu contre un adversaire hostile (et non plus "indifférent", comme dans le jeu contre la nature du chapitre III), qui implique la prévision des mouvements de l'adversaire, grâce au développement d'un arbre de possibilités. Cet arbre est généré par toutes les transitions discrètes de la situation, qu'elle vienne du système (lancement d'une observation ou d'une riposte), ou de son adversaire (apparition ou disparition d'un objet, déplacement, changement de comportement): un avion menaçant est-il seul ou fait-il partie d'une attaque concertée, une menace peu probable mais très urgente doit-elle être prise en compte, peut-on laisser sans surveillance tel secteur pendant tant de temps ?

La spécification précise de la tâche de surveillance dépend donc d'une planification prévisionnelle des ressources de surveillance, capteurs et ripostes, en fonction des possibilités d'évolution de la situation, planification qui s'apparente à la méthode du minimax de la théorie des jeux. D'une manière plus générale, même lorsqu'il s'agit d'un "jeu contre la nature", les algorithmes employés dans cette planification haut niveau relèvent des méthodes de base de résolution de problèmes en intelligence artificielle (par exemple A*, ou autre).

CHAPITRE V

SCENARIO D'INTEGRATION MULTICAPTEUR

5.1 Introduction

Cette simulation a été développée afin d'intégrer les aspects des deux précédents programmes de simulation, présentés respectivement en 2.3.1 et en 4.2, à savoir les différents types d'objets menaçants du premier avec leurs caractéristiques bidimensionnelles, les temps critiques du premier, et l'aspect probabiliste de l'intégration et du contrôle du second. Elle a été réalisée avec Jérôme de Dinechin, élève de l'ENSTA, au cours d'un stage qui a eu lieu à l'ETCA fin 1987. Nous verrons en 5.5 que nous retrouvons dans ce scénario des éléments des chapitres I et II, ainsi que les thèmes de liaisons entre modèles probabilistes et modèles intelligence artificielle développés en 4.1.1 et 4.2.6.

Il s'agissait de modéliser le comportement d'un navire muni d'une plateforme multicateur, chargé d'intercepter des missiles mer-mer, l'échec d'une interception étant sanctionné par la destruction dudit navire. La fonction de coût de surveillance utilisée en 4.2 n'est donc plus de mise, puisqu'on dispose d'une mesure directe de la performance du système dans sa tâche: sa probabilité de survie face à une attaque. On devait en outre mettre en valeur les points théoriques suivants:

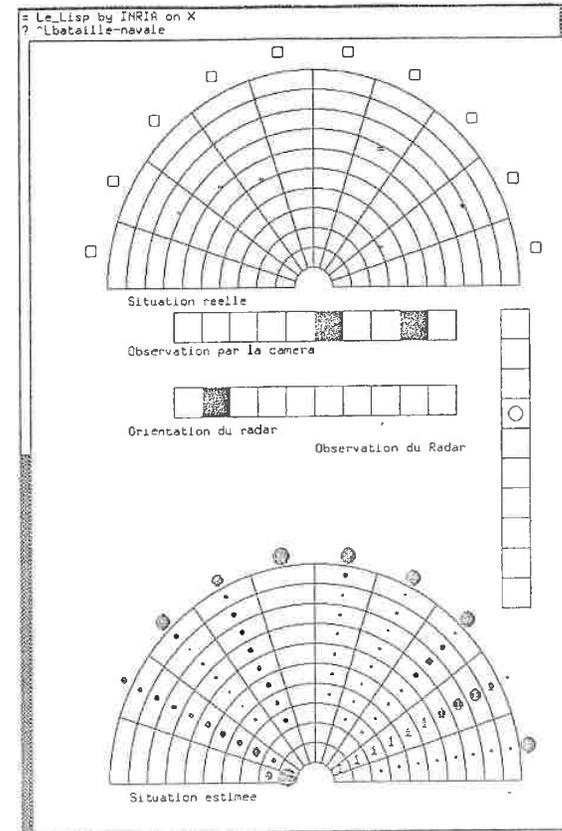
- Utilisation combinée des capteurs
- Intégration bayésienne des différentes observations en présence de bruit
- Utilisation de connaissances a priori sur le comportement des différents objets et l'évolution d'une situation.

Nous considérons un système destiné à assurer la protection d'un navire contre des missiles mer-mer arrivant vers lui au ras des flots. Ce système comprend deux capteurs, une caméra et un radar comme moyens d'observation et est doté d'un canon comme moyen d'action. Les capteurs sont modélisés très simplement. Il s'agit en premier lieu d'une caméra à grand angle, qui renvoie des informations binaires en gisement: dans tel secteur, je vois quelque chose, et dans tel autre, je ne vois rien. En second lieu, nous avons un radar, lequel ne peut couvrir qu'un petit secteur angulaire, mais qui fournit des informations en distance, vitesse, et visibilité pour chaque objet présent dans ce secteur. Nous intégrons donc ici, toujours dans le même cadre probabiliste, la perception de ces capteurs et l'évolution de la scène. Comme au chapitre IV, on observe des objets, qui peuvent être dans ce modèle de différents types: afin de ne pas compliquer démesurément les calculs, nous n'en avons retenu que quatre: les missiles lents, moyens, rapides, et les vagues.

Le système sait donc qu'un missile est un objet dangereux se dirigeant droit vers lui avec une vitesse fixe à choisir entre trois, et qu'une vague est un objet non dangereux, immobile, qui est susceptible d'apparaître et de disparaître avec des probabilités données. La présence de vagues correspond à une catégorie d'erreurs assez courantes dans la réalité, qui n'existaient pas dans le modèle présenté en 4.2, et qui ne sont pas indépendantes du temps (puisque par exemple une vague subsiste pendant plusieurs instants consécutifs, ou comme un objet non significatif dans un autre scénario de surveillance). Puisque non indépendantes du temps, ces erreurs ne peuvent être traitées par l'intégration bayésienne que si l'on a incorporé la possibilité d'existence de vague à l'espace des états.

Si l'on veut intégrer nos observations de cette manière, le choix de l'espace des états est important puisque d'une part, il faut discriminer du mieux possible les objets réels, mais que d'autre part, un cardinal trop grand de cet espace mène rapidement à une quantité de calculs rédhibitoire. Dans le cas présent, nous avons codé l'espace comme un écran radar, subdivisé en dix secteurs polaires, et chaque secteur en dix cases de longueurs égales. Nous estimerons pour chaque case la probabilité de présence d'un objet non typé, la visibilité majoritaire ainsi que le type majoritaire dans cette case. Cet artifice permet de ne pas compliquer outre mesure l'espace des états (qui serait autrement encore multiplié), tout en conservant le maximum d'information.

Dès lors, à chaque pas de temps, le navire peut mettre à jour dans cet espace sa table de densités de probabilités à partir des observations que lui renvoient ses capteurs, et des connaissances qu'il a sur les différents objets: les missiles se rapprochent du bateau en fonction de leur vitesse, les vagues ne bougent pas ou disparaissent aléatoirement. Lorsqu'une menace se fait trop pressante, il faut tirer et prendre en compte la probabilité de coup au but pour l'estimation suivante.



visualisation de la simulation

5.2 Modélisation de l'évolution de la scène et de la perception

5.2.1 Modèle d'évolution

La simulation effectuée ici doit rendre compte des deux aspects du problème. Nous devons d'abord simuler une situation réelle et la faire évoluer de manière cohérente. Mais nous devons aussi simuler la perception qu'a le navire de cette situation et l'interprétation qu'il en construit.

La situation réelle est représentée par une liste d'objets, chaque objet correspondant à un missile d'une vitesse donnée, ou à une vague. Les objets sont donc dotés dans la simulation de l'évolution de deux attributs, leur type et leur vitesse, en fonction desquels on déduit leur comportement et les observations (une vague a une taille plus importante qu'un missile, et est donc plus visible, à la fois par la caméra et le radar).

L'évolution réelle applique successivement les traitements suivants à la situation réelle:

-Si un missile est sur le point d'atteindre le navire, ce dernier est détruit. Sinon, on retranche la vitesse à la distance pour tous les objets.

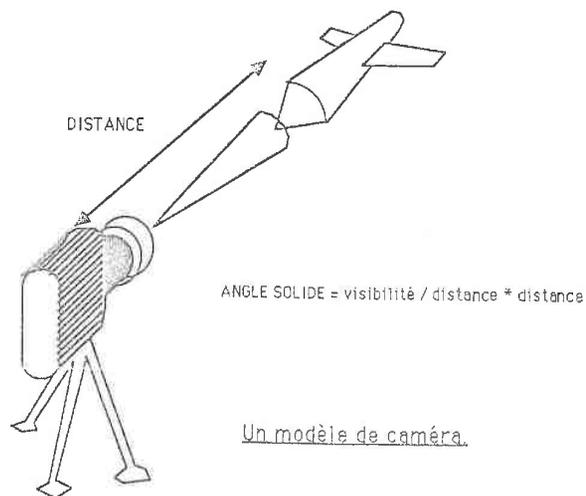
-Si le canon tire, les missiles présents dans le secteur du tir sont soumis à un tirage au sort: leur probabilité de destruction est inversement proportionnelle à leur distance du navire.

-Parallèlement, pour chaque vague, des tirages au sort décident de leur apparition ou disparition, ainsi que de leur taille apparente pour le radar ou la caméra.

-L'apparition de nouveaux missiles est déterminée soit par un scénario conçu à l'avance, soit par un pointage de la souris en mode interactif, permettant ainsi la génération à volonté d'une ou de plusieurs menaces venant de plusieurs secteurs.

5.2.2 Modèle de la perception

Une loi de comportement des capteurs étant nécessaire, nous avons modélisé l'observation de manière simple et réaliste. Attardons nous tout d'abord sur la simulation de l'observation de la caméra:



On évalue la visibilité globale de chaque secteur en additionnant sur l'ensemble des objets présents le rapport de la taille individuelle au carré de la distance du navire, à un terme "0,1" près, servant à éviter les divisions par zéro. On en déduit la probabilité de voir quelque chose par la fonction:

$$P(\text{visi-caméra}) = \min \{ (p_{\text{fa-radar}} + \text{coef-radar} * \text{visi-caméra}), 1 \}$$

La probabilité p_{fa-caméra} correspond naturellement à la probabilité de fausse alarme, puisqu'on l'obtient pour une visibilité nulle.

De même, pour simuler ce que perçoit le radar, à chaque objet présent dans le secteur on associe une liste de probabilités que cet objet soit perçu à telle ou telle distance. Un tirage au sort donne le résultat de l'observation. Il est cependant à noter que le radar est totalement précis quant à la vitesse et à la visibilité de l'objet dans notre modélisation. Pour chaque objet, on commence par calculer une nouvelle visibilité spécifique radar, qui est inversement proportionnelle à la puissance quatrième de la distance, puisqu'il s'agit ici d'un capteur actif. La probabilité conditionnelle d'observation est alors, d'une manière identique à tout à l'heure, de:

$$P(\text{visi-radar}) = \min \{ (p_{\text{fa-radar}} + \text{coef-radar} * \text{visi-radar}), 1 \}$$

On se donne ensuite des probabilités d'obtenir la distance exacte, ou la distance plus ou moins une case. Les valeurs de visibilité et de vitesse sont passées de manière exacte dans l'observation radar, mais cela ne diminue en rien le réalisme si l'on considère que la visibilité est perçue comme la taille du spot radar, et que la vitesse est très grossièrement estimée dans ce modèle.

5.3 L'estimation récursive

Nous reprenons ici les deux étapes de l'estimation récursive, détaillées au chapitre III: prédiction et intégration bayésienne des observations.

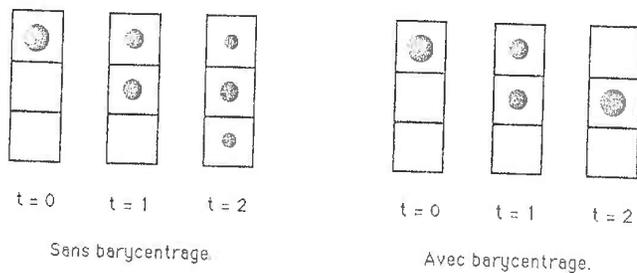
5.3.1 Prédiction

Nous avons choisi de faire évoluer le temps pas à pas, pour des raisons de simplicité des calculs. En premier lieu, nous essayons de traiter les échos liés aux vagues. D'un pas de temps au suivant, nous connaissons les valeurs de leur probabilités d'apparition et de disparition, qui sont d'ailleurs homogènes sur l'ensemble des cases dans notre modèle, et nous pouvons l'intégrer dans la prédiction. Hormis une nouvelle observation par le radar, le seul moyen pour changer d'un type rapide vers un type lent dans l'estimée a été introduit ici, en ce sens que si la probabilité d'apparition d'une vague est supérieure à la probabilité de présence existant déjà dans la case, c'est le type vague qui l'emporte. Dans le même ordre d'idée, lorsque le système tire,

nous connaissons la probabilité de coup au but pour toute case du secteur considéré, que nous pouvons également intégrer dans l'estimée.

Ensuite une prédiction sommaire prend en compte l'évolution des objets, selon la règle: "le type le plus rapide a le pas sur le type le plus lent". En effet, si la vitesse résultant de la combinaison d'un objet lent se faisant rattraper par un objet rapide, n'était pas celle de l'objet rapide, il y aurait possibilité de sous-évaluation d'une menace, ce qu'on veut éviter à tout prix. Il s'agit donc d'une approximation, puisqu'on remplace les probabilités réelles d'existence des différents types par une estimation. Pour faire évoluer la situation estimée, maintenant que sont traitées les apparitions et disparitions, le système va considérer chaque case, et propager son contenu vers le navire en proportion de sa vitesse.

Remarquons ici combien une propagation discrète est coûteuse en précision: la probabilité de présence d'un objet, une fois décentrée, ne peut plus se recentrer, à moins d'une nouvelle observation précise du radar. Une propagation des objets avec barycentrage a été envisagée, mais n'a pu être appliquée, en raison de la difficulté d'interprétation des probabilités estimées: que faire si nous avons plusieurs objets du même type dans un secteur? La solution, plus complexe serait bien sûr celle évoquée au chapitre II, d'utiliser des représentations objets pour chaque détection (dans l'estimation, et non plus seulement dans la simulation de l'évolution 5.2.1), dont on pourrait évaluer ensuite les différents attributs: type, vitesse. On retrouve là aussi une tendance naturelle à l'utilisation de ce type de représentations.



propagation d'un écho de vitesse 0.5.

5.3.2 Intégration des observations

Avant d'aborder la prise en compte des observations, il faut tenter de clarifier au maximum notre notion d'espace des états, et ce que représente chaque point dans la situation estimée. Nous avons en effet été obligé de réaliser un compromis entre les espaces des états qui seraient naturellement associés aux deux capteurs. Une remarque préalable est que tous les secteurs sont indépendants entre eux, ce qui fait que l'espace des états général sera le produit de l'espace des états dans chaque secteur. Si l'on élimine ensuite le cas de deux objets présents simultanément dans la même case, l'espace des états complet pour un secteur donné est $\{0, 1\}^n$ si n est le nombre de cases dans un secteur, 0 ou 1 signifiant l'absence ou la présence d'un objet dans la case du secteur concerné. Il faudrait donc pour cet espace des états manipuler dans un seul secteur 2^n probabilités, nombre qui explose rapidement et qui est impraticable. De la même manière que dans le précédent modèle, nous allons donc nous ramener à une approximation, en profitant du caractère local de l'évolution des objets et de la perception.

Plusieurs choix sont alors possibles, selon qu'on aborde le problème sous l'angle de la caméra ou celui du radar. Tout d'abord, pour la caméra, la notion de base est qu'il y a ou non un objet dans un secteur. Aucune idée de distance n'intervient donc et en faisant jouer le masquage des objets les uns par rapport aux autres, chaque case occupée recouvre en fait les cases qui sont derrière elle.

L'espace des états se formaliserait donc de la manière suivante: à chaque case est associée la probabilité que le premier objet en vue soit à cette position et n'importe quoi aux positions plus éloignées du même secteur. Cependant, une notion de ce type pourrait faire qu'un objet inoffensif masque un objet offensif, et soit cause de la destruction du navire. Notre caméra, d'ailleurs, ne prend pas en compte le masquage des objets. L'avantage d'une telle interprétation réside dans sa simplicité de mise en oeuvre et de compréhension, pour ce qui est de l'intégration des résultats d'observation de la caméra. L'espace des états complets (pour un secteur) serait constitué des n états déterminés par la présence ou l'absence du premier objet dans chacune des n cases.

état 0: $sc_0 = (1 \ x \ x \ x \ \dots \ x)$
 état 1: $sc_1 = (0 \ 1 \ x \ x \ \dots \ x)$
 ..
 état $n-1$: $sc_{n-1} = (0 \ 0 \ 0 \ \dots \ 0 \ 1)$
 état n : $sc_n = (0 \ 0 \ 0 \ \dots \ 0 \ 0)$

X pouvant être indifféremment 0 ou 1. La somme de la probabilité de tous ces états est égale à 1, celle du dernier état représentant la probabilité qu'il n'y ait pas d'objets dans le secteur concerné dans la limite de la portée de la perception, donc des n cases. On considère ici implicitement que les objets les plus proches sont les plus visibles, et donc sont dominants pour le résultat de l'observation par la caméra.

Abordons maintenant le problème sous l'angle radar. Il faut que nous ayons la possibilité d'avoir plusieurs objets bien discriminés en distance dans un même secteur. On peut penser à mettre dans chaque case la probabilité de présence d'un objet à la distance correspondante, indépendamment de la présence d'objet dans les autres cases. Plus précisément, il nous faut comme dans le précédent modèle, approximer la probabilité d'un état quelconque $(s_i)_{i=0, \dots, n-1}$ par le produit des probabilités de présence ($s_i=1$) ou d'absence ($s_i=0$) élémentaires en chaque case i .

état 0: $s_0 = (1 \ X \ X \ \dots \ X)$ complémentaire non- $s_0 = (0 \ X \ X \ \dots \ X)$
 état 1: $s_1 = (X \ 1 \ X \ \dots \ X)$ non- $s_1 = (X \ 0 \ X \ \dots \ X)$
 ...
 état $n-1$: $s_{n-1} = (X \ X \ \dots \ X \ 1)$ non- $s_{n-1} = (X \ X \ \dots \ X \ 0)$

C'est cette représentation approchée que nous avons choisie, puisqu'elle va nous permettre d'intégrer à la fois les observations caméra et radar. Ici, les probabilités sont normalisées deux à deux, puisque $P(s_i) + P(\text{non-}s_i) = 1$.

Le point crucial consistera, bien sûr, en la normalisation de la loi de probabilité sur l'ensemble des états après l'intégration bayésienne, puis le retour à la forme initiale des vecteurs de probabilité. Rappelons nous, en effet, le processus d'intégration bayésienne.

$$P_{\text{a-posteriori}}(X) = P(O|X) * P_{\text{a-priori}}(X) / (\sum_i P(O|X_i) * P_{\text{a-priori}}(X_i))$$

Nous ne connaissons rien de $P(O)$, ce qui oblige à l'éliminer par normalisation sur l'ensemble des états possibles. Or, X_i doit décrire l'ensemble complet des états, c'est-à-dire les 2^n possibilités par secteur. Alors que tout-à-l'heure, l'intégration de l'observation caméra était facile à opérer, puisque chacun de nos états grossiers avait la même visibilité, mais l'observation radar était dure à mettre en place, nous ne connaissons maintenant que quelques extraits d'états, et la normalisation ne s'opère plus aussi facilement. Si s est le vecteur composé de zéros et d'unités représentant un état complexe, et si la case j de la situation estimée représente la probabilité p_j de présence d'un objet à cette distance sans rien supposer sur le reste du secteur, on écrit:

$$P(s) = \prod_j [s_j * p_j + (1 - s_j) * (1 - p_j)]$$

en supposant que les objets présents aux différentes distances soient indépendants.

Donc pour un secteur donné (puisque'ils sont tous indépendants), la densité de probabilité des états est donnée par un vecteur de probabilité de taille n , d'une taille polynômiale, et non exponentielle, comme l'exigerait la prise en compte de tous les états. La coordonnée d'indice i représente la probabilité qu'il y ait un objet dans le secteur considéré à la case i , indépendamment des autres cases, et n'a a priori rien à voir avec la somme des autres coordonnées.

Voyons ce que donne l'intégration bayésienne de l'observation pour la caméra. La mise à jour de la situation estimée à partir de l'observation faite par la caméra est effectuée grâce à notre approximation de l'erreur liée à cet instrument et à notre estimation avant l'observation en utilisant la loi de Bayes. A partir de la représentation utilisée, il s'agit de repasser dans celle permettant l'intégration de l'observation caméra. Cela se fait simplement, puisque:

$$\begin{aligned} P(sc_0) &= P(s_0) \\ P(sc_1) &= (1-P(s_0)) P(s_1) \\ &\dots \\ P(sc_{n-1}) &= \prod_{i=0}^{n-2} (1-P(s_i)) P(s_n) \\ P(sc_n) &= \prod_{i=0}^{n-1} (1-P(s_i)) \end{aligned}$$

L'intégration se fait alors aisément sous cette forme:

$$P'(sc_i) = P(sc_i) P(O|sc_i) / [\sum_{j=0}^n P(sc_j) P(O|sc_j)]$$

puisque la probabilité d'observation ne dépend que de la visibilité, le ' signifiant la nouvelle valeur de la probabilité après observation.

On peut alors se ramener à la forme standard:

$$\begin{aligned} P'(s_0) &= P'(sc_0) \\ P'(s_1) &= P'(sc_1) + \mu P'(sc_0) \\ &\dots \\ P'(s_{n-1}) &= P'(sc_{n-1}) + \mu \sum_{i=0}^{n-2} P'(sc_i) \end{aligned}$$

Bien sûr, on a là encore "oublié" une partie de l'information, puisqu'il aurait fallu décomposer suivant les 2^n états de base. C'est pourquoi on est obligé d'introduire le paramètre μ , qui prend en compte la probabilité qu'un objet soit masqué, ou qui peut être vu comme la densité moyenne des objets. Comme de toute façon celle-ci est faible, l'approximation peut être considérée comme valide.

Plus encore que précédemment, l'actualisation due à la perception radar est essentielle à l'estimation. Elle s'effectue de la façon suivante: pour chaque case de notre estimée, on va évaluer la probabilité de présence due à chacun des objets observés, et lui attribuer le type de

l'objet dont la contribution est la plus grande.

$$P'_i = P(\text{obs}|s_i) * P_i / [P_i P(\text{obs}|s_i) + (1-P_i) P(\text{obs}|non-s_i)]$$

Le problème est ici de calculer les probabilités d'observation dans les deux cas, présence ou absence d'un objet en i . S'il y a présence (état s_i), on a vu plus haut la forme de la distribution de probabilité d'observation. S'il y a absence, il faut tenir compte d'une probabilité intrinsèque de fausse alarme, ainsi que d'une éventuelle contribution d'un objet dans une autre case. Cette dernière influence reste cependant localisée.

5.3.3 Contrôle

Cette fonction est destinée à orienter à la fois le radar et le canon, de manière à optimiser les performances du système. Le contrôle du canon doit tout d'abord extraire de la table des probabilités les différentes possibilités de menace. D'une manière générale, dans un cas critique, des systèmes concurrents seraient discriminés par le choix qu'ils feraient entre une menace lointaine mais précise, et une menace proche mais floue ou peu probable. On verra en 5.5 comment on pourrait développer complètement le contrôle dans ce sens en utilisant les techniques de la théorie des jeux. Diverses options pourraient faire en outre faire intervenir explicitement dans la fonction de coût une limitation du nombre des obus. Ici nous nous sommes limités à un tir par pas de temps et à un système de contrôle simple dérivé de celui utilisé en 4.2. La modélisation évalue la menace en sommant dans chaque secteur la densité de probabilité de présence d'un missile, sans tenir compte de son type, pondérée par un facteur de distance. Si la valeur obtenue dépasse un certain seuil, le navire tire. On a donc un contrôle de la sensibilité du système par ce seuil.

L'orientation du radar influe également de manière décisive sur les performances du système, et est peut-être même encore plus importante, puisque le contrôle du canon n'entre en jeu qu'au-delà d'une certaine précision de la menace. Il s'agit en effet de déterminer les zones les plus incertaines, déjà délimitées par la caméra, pour aller y regarder de plus près avec le radar. D'autre part, par opposition avec le modèle de contrôle d'une fenêtre d'observation en 4.2, il ne s'agit plus ici d'une surveillance stricte, où on optimise simplement la connaissance sur la scène, mais d'un bouclage sur la réalité puisqu'on cherche à maximiser le taux de destruction des missiles, à minimiser la consommation de munitions, plus d'autres éventuels critères moins significatifs. Il faut donc introduire dans le contrôle du radar des éléments sur l'importance de la détection des missiles, et ce d'autant plus qu'ils sont plus proches et plus rapides.

A cet effet, nous avons testé différentes stratégies pour l'orientation du radar, qui incluent séparément les différents éléments d'appréciation de la situation que nous venons de présenter, afin d'avoir une idée de leur influence qualitative. Nous ne les avons pas testé en continu comme en 4.2, d'autant qu'ici les simulations se concluent rapidement en faveur du bateau ou des

attaquants, mais plutôt face à des situations types: 2 missiles arrivant dans deux secteurs différents, 3 missiles dont 2 dans le même secteur, 3 missiles dans des secteurs différents. Pour chaque scénario, nous avons répété l'"expérience" un millier de fois pour avoir une bonne mesure des moyennes des critères de performances des différentes stratégies.

5.4 Résultats

Le code écrit tourne actuellement de manière satisfaisante. Plusieurs combinaisons des fonctions de contrôle du radar et du canon ont été testées, puisque la mesure de performance porte sur l'ensemble.

stratégies d'orientation du radar:

- r0: oriente simplement le radar vers la détection la plus forte. Cette stratégie, qui n'utilise aucune optimisation, correspond à une poursuite classique qui marche bien lorsqu'il n'y a qu'un seul missile, et sert de point de référence pour les comparaisons.
- r1: cette stratégie oriente le radar vers le secteur qui présente le plus de probabilités de détection. Elle incorpore donc l'intérêt du système pour la détection des missiles, tout en étant moins simpliste que la précédente. Par contre, elle n'incorpore pas le fait de s'orienter sur le secteur d'incertitude maximale, comme la stratégie utilisée en 4.2. Le même effet est quand même réalisé implicitement, dans la mesure où il n'y a que peu de missiles, puisqu'un coup de radar précipite les probabilités en ne laissant aucune ou une seule case avec une forte probabilité, diminuant ainsi la somme des probabilités sur le secteur. Le coup suivant observera donc probablement un autre secteur.
- r2: cette stratégie reproduit celle de 4.2, qui oriente le radar vers le secteur présentant l'incertitude maximale. Elle n'incorpore pas l'importance de la détection des missiles (pour elle savoir qu'il y a un missile est aussi important que de savoir qu'il n'y en a pas), ni celle de la proximité.
- r3: cette stratégie pondère l'incertitude en fonction de la proximité des cases.

Les tableaux qui suivent présentent les résultats de ces différentes techniques d'orientation du radar. Seul le seuil de sensibilité c permet de moduler la stratégie de contrôle du tir. Outre les deux critères déjà vus: taux de succès dans la destruction des missiles, et consommation de munitions, nous avons introduit les temps moyen de destruction du missile en cas de succès, et de destruction du navire en cas d'échec. Ils permettent également de qualifier la stratégie utilisée, qui sera d'autant meilleure que le temps moyen de succès est court, et que le temps moyen d'échec est long.

scénario 1: deux missiles dans deux secteurs différents

	taux de succès	munitions	temps moyen succès	temps moyen échec
r0 + c0,1	0,42	5,91	6,43	12,16
r0 + c0,2	0,39	4,38	6,30	12,92
r0 + c0,3	0,37	3,57	6,19	13,40
r0 + c0,05	0,59	7,60	7,67	8,74
r1 + c0,1	0,55	6,00	7,93	9,54
r1 + c0,2	0,63	5,35	9,89	7,76
r1 + c0,3	0,59	4,25	9,73	8,58
r2 + c0,1	0,61	7,37	8,33	8,13
r2 + c0,2	0,62	5,80	9,94	8,08
r2 + c0,3	0,53	4,21	8,83	9,85
r2 + c0,4	0,47	3,27	8,09	11,17
r3 + c0,1	0,57	6,97	7,60	9,11
r3 + c0,2	0,52	5,21	7,81	10,13
r3 + c0,3	0,52	3,85	8,32	10,09
r3 + c0,4	0,55	3,54	9,45	9,53

scénario 2: trois missiles dont deux dans le même secteur

	taux de succès	munitions	temps moyen succès	temps moyen échec
r0 + c0,1	0,40	7,86	6,97	12,56
r0 + c0,2	0,38	6,53	7,58	12,97
r0 + c0,3	0,33	5,19	6,63	14,13
r1 + c0,05	0,64	10,64	10,96	7,64
r1 + c0,1	0,61	8,65	10,92	8,22
r1 + c0,2	0,62	7,23	11,80	8,11
r1 + c0,3	0,61	6,01	12,20	8,24
r2 + c0,1	0,61	9,47	10,08	8,27
r2 + c0,2	0,52	7,69	9,67	10,12
r2 + c0,3	0,52	6,10	10,34	10,08
r2 + c0,4	0,42	4,53	8,84	12,18
r3 + c0,1	0,55	9,83	8,97	9,45
r3 + c0,2	0,52	7,76	9,61	10,22
r3 + c0,3	0,50	5,92	9,49	10,59
r3 + c0,4	0,39	4,37	7,86	12,92

scénario 3: trois missiles dans trois secteurs différents

	taux de succès	munitions	temps moyen succès	temps moyen échec
r0 + c0,1	0,23	7,72	4,66	18,11
r0 + c0,2	0,16	5,18	3,34	19,47
r0 + c0,3	0,11	4,00	2,53	20,22
r1 + c0,05	0,30	10,93	5,77	17,76
r1 + c0,1	0,20	8,42	4,01	19,74
r1 + c0,2	0,33	7,15	6,98	16,35
r1 + c0,3	0,28	5,39	6,23	17,94
r2 + c0,1	0,26	8,98	5,05	18,37
r2 + c0,2	0,27	6,48	5,58	18,00
r2 + c0,3	0,21	5,04	4,78	19,13
r2 + c0,4	0,16	3,68	3,82	20,22
r3 + c0,1	0,25	8,72	5,07	18,22
r3 + c0,2	0,19	6,41	4,13	19,54
r3 + c0,3	0,15	4,88	3,52	20,16
r3 + c0,4	0,14	3,52	3,27	20,33

On observe dans un premier temps que l'effet de baisser le seuil de sensibilité c du tir est bien d'augmenter le nombre de munitions consommées. Par contre, cela n'augmente pas toujours le taux de survie, cela étant probablement dû à la prise en compte dans la prédiction de l'effet de ce tir. Comme prévu, la stratégie $r0$ est inefficace sur des scénarios incluant plusieurs missiles, la meilleure du point de vue des taux de succès étant $r1$, avec un seuil c de 0,2. En fait, il n'y a pas une technique réellement indiscutable, ce qui n'est pas étonnant, puisque nous n'avons pas cherché à intégrer tous les aspects en une seule stratégie, ni à optimiser absolument le contrôle. Nous avons par exemple pris un critère composite incluant le taux de succès et la consommation:

	perfo = 1+ taux succès - 1/10 * munitions
r3 + c0,4	1,1929
r1 + c0,3	1,1674
r2 + c0,4	1,1431
r3 + c0,3	1,1349
r2 + c0,3	1,1114
r1 + c0,2	1,0972
r2 + c0,2	1,0365
r0 + c0,3	1,0083
r3 + c0,2	0,9994
r0 + c0,2	0,9526
r1 + c0,1	0,9463
r2 + c0,1	0,8762
r3 + c0,1	0,8599

Ce critère nous donne donc un exemple d'optimisation du contrôle. Le fait que la stratégie r1 n'inclue pas de mesure de l'incertitude n'est pas forcément surprenant, si l'on prend en compte ce que nous avons vu plus haut sur l'effet d'une mesure radar (alors que la stratégie r0 a tendance à rester accrochée sur un secteur). D'autre part, la stratégie de contrôle du canon relativement simple utilisée, par opposition avec celle exposée en 5.5, ne permet peut être pas de juger complètement la partie orientation du radar. Néanmoins les performances de r2 sont assez robustes par rapport à la sensibilité du canon, et la stratégie r3 donne un bon résultat, en particulier si l'on accroît l'importance de la consommation dans le critère composite:

	perfo = 1+ taux succès - 1/4 * munitions
r3 + c0,4	0,6616
r2 + c0,4	0,6529
r3 + c0,3	0,5572
r1 + c0,3	0,5305
r2 + c0,3	0,4792
r0 + c0,3	0,4733
r0 + c0,2	0,2964
r1 + c0,2	0,2947
r3 + c0,2	0,2177
r2 + c0,2	0,1659
r1 + c0,1	0,0467
r0 + c0,1	-0,0557
r3 + c0,1	-0,1759
r2 + c0,1	-0,2299
r1 + c0,05	-0,3105

Naturellement, plus on accroît l'importance de la consommation en munitions, plus les stratégies sont regroupées suivant le seuil de sensibilité, et moins le contrôle radar a d'importance. Nous n'avons malheureusement pas pu explorer complètement toutes les différentes possibilités de combinaison, compte tenu de la complexité de l'ensemble: outre d'autres stratégies pour le canon, on pourrait également utiliser les indications de type et de vitesse. On peut cependant noter une amélioration nette des performances, de 40 à 60% de taux de succès, de la stratégie r0 la plus simple, aux stratégies prenant en compte le contrôle de la perception et les probabilités des différentes hypothèses. Le choix de la combinaison adéquate dépend donc du critère retenu pour le système. Là encore, et d'une manière beaucoup plus nette que le modèle présenté en 4.2, on voit la nécessité d'un système symbolique fixant le choix de la méthode d'optimisation, en fonction de données internes (quantité de munitions disponibles) et externes sur la situation (évaluation de la menace qualitativement et quantitativement, directions les plus probables, état de la mer influant sur la perception des missiles). Nous allons donc développer cet aspect dans la partie suivante.

5.5 Liens avec l'intelligence artificielle

5.5.1 Représentations utilisées

Par rapport aux exemples du chapitre IV, ce modèle incorpore plus de types objet dans l'espace des états, tout en restant assez simple. Un traitement plus complet que celui réalisé ici traiterait ces objets en tant que tels, comme nous l'avons vu plus haut. Si nous récapitulons ces représentations, nous trouvons pour l'environnement des missiles, caractérisés par leur vitesse (lente, moyenne, rapide), ou des vagues génératrices de fausses alarmes. D'autre part, la perception comprend deux capteurs, une caméra et un radar, la première couvrant tout le champ, le dernier n'observant qu'un secteur et devant être orienté.

Si maintenant nous envisageons un modèle à deux couches, analogue à celui exposé en 4.2.6 (haut niveau intelligent ↔ estimation et contrôle probabiliste), on peut également envisager dans le haut niveau des analyses globales et qualitatives de la situation:

- détection et traitement de salves de missiles (notion de groupe d'objets)
- classification qualitative de la situation en différents contextes: veille, menace, situation d'urgence

Enfin, une information importante sur le système qui n'est pas intégrée dans la simulation actuelle, devrait également être prise en compte dans cette couche "intelligente": la quantité de munitions disponibles, qui influe d'une manière évidente sur la stratégie de défense et de riposte (et par extension, la disponibilité des réserves en énergie..).

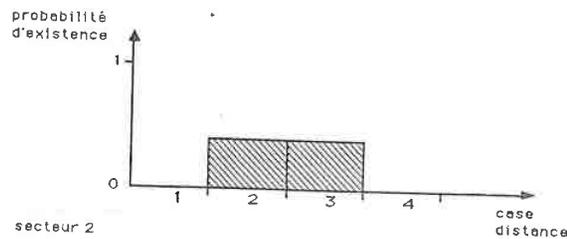
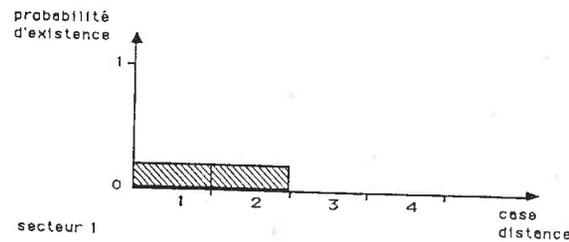
5.5.2 Approximations dans le modèle probabiliste et développement de l'arbre des hypothèses

Nous avons déjà parlé au chapitre III de la robustesse des modèles probabilistes. C'est cette robustesse et cette tolérance que nous mettons à profit lorsque nous faisons des approximations dans l'espace des états du modèle, au prix d'une perte d'information à chaque étape de prédiction ou d'intégration des observations (dans cette simulation-ci, et en 4.2) ou dans les calculs de probabilité eux-même, au prix d'une perte sur leur précision (ici par exemple lorsqu'on attribue à un objet le type majoritaire, au lieu d'évaluer en parallèle toutes les hypothèses).

D'une manière intéressante, ces approximations conduisent souvent simultanément à un rapprochement avec les méthodes symboliques utilisées en intelligence artificielle, soit que les espaces des états simplifiés conduisent à une description "naturelle" du modèle (par exemple dans notre simulation, ou en 4.2, on a ainsi une attribution directe d'une probabilité d'existence d'objet à une case d'espace, ce qui ne serait pas le cas dans un traitement global de l'espace, comme le voudrait l'application stricte de la théorie), soit que les approximations dans l'estimation recréent une description en plusieurs niveaux de la scène et des objets, puisque

l'estimation devient hiérarchisée, comme dans le cas cité ci-dessus de l'attribution du type de l'objet.

Enfin, toujours de la même manière qu'en 4.2.6, la prédiction nécessitée par la recherche de la stratégie optimale conduit, si l'on veut traiter complètement le problème (ce qui n'est pas le cas ici, puisqu'on se contente comme en 4.2 d'une fonction indice pour les contrôles du radar et du canon), à un développement de l'arbre des hypothèses engendré par les transitions discrètes. La détermination des meilleures commandes pour les radar et le canon utilise alors des méthodes inspirées de celles de la théorie des jeux, de la recherche opérationnelle et de l'intelligence artificielle (parcours d'arbre, minimisation de distance dans un graphe, recherche du chemin critique). Pour prendre un exemple simple: imaginons que deux menaces aient été détectées dans deux secteurs différents, l'une peu probable mais urgente (c'est à dire laissant peu de délai de réaction), l'autre plus probable et moins urgente.

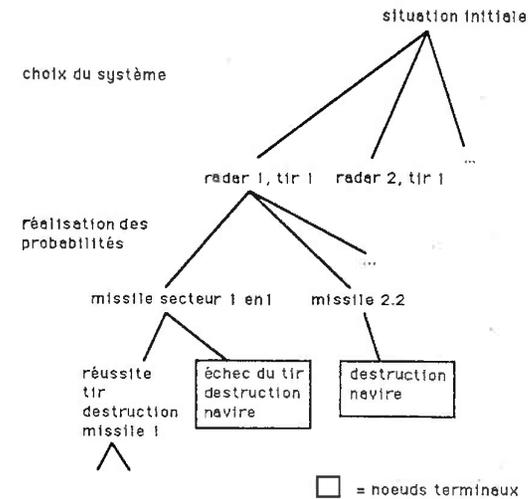


description du tableau des probabilités dans les deux secteurs

Dans ce cas élémentaire, en prenant par exemple pour les menaces dans les deux secteurs la même vitesse de 1, les choix qui s'offrent au système sont à chaque instant les suivants:

- pointage du radar dans l'une ou l'autre direction, ce qui a pour effet la "réduction du paquet d'onde", c'est à dire que la position du missile est précisément déterminée, ainsi que son existence avec une probabilité finale proche de 0 ou proche de 1
- tir du canon dans un des deux secteurs avec une certaine probabilité de succès s'il y a bien un missile, ou absence de tir.

On est donc conduit à développer un arbre à partir des probabilités de base vues dans le tableau ci-dessus, ainsi que des différentes possibilités de choix précédentes:



développement de l'arbre des hypothèses

Nous devons déjà étudier, même dans ce cas très simple (et un peu critique !) un arbre assez complexe, mais pratiquement identique à ceux développés lors des jeux à deux joueurs (jeu d'échecs ou autre), et dans laquelle les choix optimaux sont déterminés par la méthode de décision du minimax (critère de décision du minimax vu au chapitre III appliqué récursivement sur l'arbre en remontant les valeurs d'une fonction de coût sur les feuilles de l'arbre, alternativement par le min (choix du joueur) et par le max (choix de l'adversaire) sur les nœuds). La différence ici est que certaines branches sont étiquetées par des probabilités, ce qui veut dire que la stratégie de l'adversaire est en principe connue a priori. Dans le cas d'un "jeu contre la nature", correspondant à un adversaire non hostile, on peut alors simplifier la méthode de prise de décision du minimax en remplaçant l'opération max (maximisation par l'adversaire de la fonction de coût du système) par un simple calcul d'espérance, en pondérant les coûts avec les probabilités des différentes hypothèses.

Dans le cas d'un adversaire intelligent et hostile, certaines informations a priori, telles que la probabilité non négligeable d'occurrence d'une salve de missiles, dans des secteurs différents, ou dans le même secteur (possibilité éliminée dans l'arbre ci-dessus, mais qui peut aisément être prise en compte), ne sont pas incorporées dans le modèle, puisqu'on a localisé les probabilités

d'existence des objets. Il est alors plus prudent de conserver le critère du minimax, en pondérant les risques ou les coûts par les probabilités. Encore une fois, il est caractéristique de constater ici que l'utilisation des méthodes d'intelligence artificielle est d'autant plus nécessaire que le modèle a été simplifié, c'est à dire que l'on a remplacé un calcul probabiliste par un calcul symbolique, et ce d'autant plus que, contrairement au cas "jouet" que nous venons de développer, la taille des problèmes réels (ou même simplement du problème anti-missile de ce chapitre avec tous les secteurs, et toutes les cases remplies par des probabilités non nulles) impose pratiquement le recours à des heuristiques, à des règles expertes du domaine, afin de permettre l'élagage de cet arbre explosif, et le choix d'une bonne stratégie.

5.5.3 Liaisons avec un système d'intelligence artificielle

Nous venons de nous rendre compte que certaines informations importantes ne sont pas intégrées dans le modèle probabiliste, outre le fait que ce même modèle a besoin pour fonctionner de probabilités a priori et d'indications stratégiques. Tout ce qui a été évoqué en 4.2.6 peut être repris ici, et même développé du fait de la complexité supérieure du modèle, et du plus grand nombre d'informations a priori et externes à intégrer. On retrouve alors l'intérêt d'une couche de haut niveau capable d'intégrer les directions probables des menaces, l'analyse du comportement global des objets, la classification de la situation dans un contexte qualitatif, ou même une partie planification indiquant une direction stratégique de la tâche en fonction de la situation. Ce niveau fournit au modèle probabiliste les probabilités a priori d'existence et d'apparition en fonction de l'analyse de la situation et des objets détectés.

La stratégie du système peut alors elle aussi être déterminée dans cette couche "intelligente" en fonction de critères tels que l'intensité de la menace dans la région, la durée de la mission, ... le principal paramètre pouvant être réglé à haut niveau étant la sensibilité du système, ou sa propension à tirer. Il peut ainsi se déclencher sur des fausses alarmes, ou choisir d'attendre que la menace se précise, au risque de perdre du temps et de gaspiller sa ressource perceptive radar. Le système doit alors prendre en compte le niveau des munitions dans la détermination de cette stratégie. Là encore, des règles heuristiques peuvent être utilisées, en liaison avec des éléments numériques. Par exemple, on pourrait se fixer un seuil, pour une diminution minimale de la probabilité de destruction du navire "achetée" par un obus, seuil calculé en optimisant l'utilisation des munitions sur la durée de la mission; on rejoint là aussi la théorie des jeux, et la recherche opérationnelle ou l'économie.

Si nous récapitulons maintenant les trois parties d'un système robotique de perception que nous avons développé à travers ces trois exemples, nous obtenons:

- une couche haut niveau d'analyse et de classification qualitative de la situation
- un module d'estimation récursive, à partir d'un modèle probabiliste basé sur les probabilités a priori fournies par la partie précédente
- un module de contrôle, utilisant à la fois les probabilités fournies par l'estimation, la fonction

de coût fournie par le haut niveau, et des techniques mixtes probabilistes/symboliques de minimisation, recherche de chemins critiques.. sur des arbres et des graphes

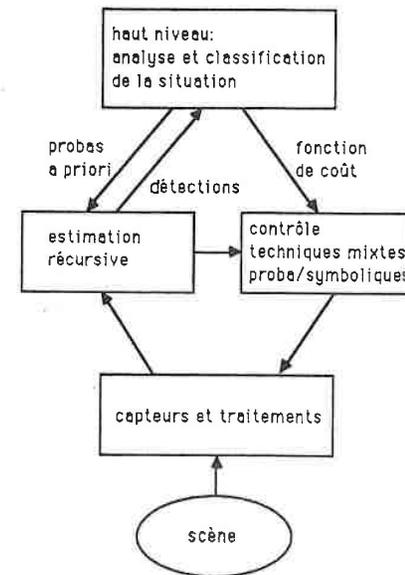


schéma d'un système de perception robotique en deux couches

VI CONCLUSION

6.1 Problème d'intégration et de contrôle de la perception

Nous pensons donc avoir montré que ce problème d'intégration et de contrôle de la perception est un problème réel, même s'il est relativement nouveau, en raison du développement de systèmes complexes de surveillance, notamment dans le domaine militaire, et de systèmes robotiques avancés. L'intégration multisensorielle ou multitraitement, et le contrôle et l'allocation des ressources perceptives telles que fenêtres d'observation capteur, capacités de traitement d'un calculateur universel ou d'un processeur spécialisé dans un système de perception temps réel, commencent à devenir nécessaires.

Nous arrivons ainsi à des méthodes d'intégration et de contrôle perceptifs permettant de réaliser des boucles de perception de bas niveau (veille, poursuite...), dans le cadre d'un modèle automatique, dans lequel des connaissances complexes sur les objets observés et leur évolution sont prises en compte, notamment à travers des représentations discrètes et des modélisations poissonniennes des transitions, comme dans les exemples présentés au chapitre IV. Ces techniques peuvent être trompées par des situations imprévues sortant de leur compétence, mais sont rapides et efficaces. La hiérarchisation des boucles dans notre modèle permet de prendre en compte ces cas imprévus, ainsi que la présence d'un système de haut niveau utilisant des techniques d'intelligence artificielle.

Nous avons présenté deux simulations de contrôle de fenêtre d'observation capteur, en 4.2 et au chapitre V. La première, dont la simplicité nous a permis de faire une étude assez complète du comportement du modèle d'optimisation de la commande du capteur, correspond déjà presque à un problème de traitement du signal: il suffirait d'intégrer des objets émettant des signaux multiniveaux pour être proche d'un problème concret de sonar ou de détection de signaux dans le spectre électromagnétique, ce qui permettrait éventuellement d'améliorer les performances de ce genre de système.

Ce travail devrait trouver une application immédiate dans la réalisation d'un logiciel de contrôle de la plate-forme multicapteur "Percept" dont va se doter l'ETCA d'ici à la fin 1988. Cette plate-forme sera dotée d'une antenne acoustique, ainsi que de deux caméras et d'un télémètre orientables. Tous ces capteurs sont commandables ou paramétrables (orientation, et zoom et mise au point pour les caméras). Les problèmes d'intégration et de contrôle de la perception ne devraient donc pas manquer... Le premier scénario étudié concerne le traitement simultané de différentes tâches (régions à observer, événements à classifier) par les capteurs, à l'aide d'un modèle d'allocation des ressources dans un système stochastique, scénario qui devrait évoluer vers la détection et la poursuite multisensorielle (vision et acoustique) d'objets mobiles susceptibles de subir des masquages, des variations de comportement, éventuellement discriminables par des traitements de reconnaissance des formes.

D'autre part, dans le domaine étudié, les progrès des systèmes militaires de détection et de poursuite devraient donner lieu à des développements intéressants de ces techniques: en balayage électronique radar ou sonar notamment, l'agilité du faisceau devrait permettre de nouvelles stratégies de contrôle, tant pour la veille que pour la poursuite multicibles. De plus, les capteurs utilisés dans le domaine militaire renvoient des images de plus en plus complexes (images radar complexes, caméras infra-rouge) analogues à celles obtenues dans les systèmes de vision, rendant nécessaires des représentations de plus en plus complexes de l'environnement (reconnaissance de clutter, présence d'objets non significatifs, plusieurs types d'objet en présence).

Dans ce domaine de la vision, des progrès en matière de traitement, et en intégration multitraitement sont à attendre. On sait déjà réaliser des détecteurs de contours optimaux, et le problème de la segmentation redondante, résistant au bruit d'observation est un thème de recherche très important, qui devrait permettre de fournir des outils de traitement visuel de base pour des robots autonomes. Pour les systèmes temps réel, la technique des fenêtres de traitement dans lesquelles on limite les traitements coûteux en temps de calcul, et positionnées par des indices préattentifs d'acquisition plus rapides, relève tout à fait du problème de contrôle d'une fenêtre d'observation étudié dans ce mémoire.

6.2 Liaison entre un système probabiliste et un système d'intelligence artificielle

Nous avons choisi la modélisation probabiliste comme moyen d'intégration des différents traitements et des informations a priori, compte tenu de l'incertitude des résultats renvoyés par les traitements du signal et les traitements d'images. Cette incertitude peut être due aux bruits et aux conditions d'observations, aux erreurs de traitements, que ce soit pour des raisons de difficultés inhérentes au problème ou pour des raisons de temps réel (par exemple en vision, traitements frustes et rapides renvoyant des indices de vision précoce ou préattentive). De plus, la présence de modèles mathématiques précis (géométriques, fréquentiels) d'évolution et d'observation du système valide encore ce choix, par rapport aux techniques de flou utilisées dans les modélisations mal définies des systèmes experts (fusion de règles expertes ...).

Les techniques probabilistes que nous avons étudiées permettent d'interfacer un système de décision intelligent travaillant sur des objets informatiques d'un haut niveau sémantique, c'est à dire directement significatifs pour la tâche à effectuer, avec les résultats issus des divers traitements appliqués aux capteurs. Cet interfaçage n'est bien sûr pas toujours indispensable, et une estimation récursive tournera sans problème tant que son modèle reste valide. Mais dans certains cas, tel que celui déjà cité d'objets en formation, si cette notion n'est pas prise en compte dans l'espace des états, une analyse de haut niveau est utile, et même nécessaire dans le cas d'une menace militaire. En fonction de cette analyse, la tâche perceptive peut être alors redéfinie: par exemple, examiner juste un des objets en détail, ou examiner tous les objets de manière homogène.

Bien sûr, nous nous limitons dans cette interfaçage aux représentations précises (au sens du chapitre II) s'intégrant dans un ensemble bien délimité d'hypothèses, les connaissances vagues, ou celles qui n'ont pu être intégrées au modèle probabilistes (contextes, groupes d'objets vu ci-dessus), relevant justement du système intelligent. Afin de réaliser cette interface, les représentations et les modèles d'évolution des objets observés, et de l'environnement doivent en effet être intégrés dans l'espace des états: objets définis non seulement par leur position et leur vitesse, mais aussi par des attributs symboliques pouvant être identifiés par des algorithmes de reconnaissance des formes en vision, en acoustique, ou notion de contexte permettant de délimiter qualitativement un ensemble de situations.

Nous avons vu, à travers l'évolution des systèmes de poursuite multicible (3.3), qu'une complexification des modèles de poursuite classique du genre filtre de Kalman intervient ainsi naturellement avec la complexification de l'univers observé, la multiplication des objets... Nous avons nous-même présenté une extension du filtre de Kalman (4.1) intégrant plusieurs hypothèses, ainsi que des caractéristiques discrètes dans l'évolution du modèle (sauts poissonniens entre différents modèles linéaires-gaussiens) et les observations (présence ou absence d'un objet, résultats de traitements de reconnaissance des formes...). Dans cet algorithme, une notion simple de contexte est intégrée dans le modèle probabiliste, sans qu'il soit forcément nécessaire de faire appel à une analyse de haut niveau externe. Ce modèle d'estimation récursive pourrait être intéressant pour une application robotique de surveillance, et nous permet d'introduire un second type de convergence avec les techniques et représentations d'intelligence artificielle, interne cette fois-ci, par rapport à la méthode externe d'une supervision intelligente présentée au paragraphe précédent.

La modélisation nécessaire d'environnements de plus en plus complexes passe en effet par des techniques probabilistes de plus en plus élaborées, relevant en particulier du domaine du filtrage non linéaire, dont ces modèles particulièrement importants mêlant des aspects discrets et continus, qui peuvent conduire à l'utilisation d'algorithmes d'allocation de ressources dans un système stochastique (4.2), ou au développement d'arbres de branchement d'hypothèses analogues à ceux de la théorie des jeux (chapitre V), amenant ainsi l'utilisation de techniques symboliques dans l'estimation et le contrôle probabiliste. Des techniques de calcul et d'approximation de ces modèles sont également nécessaires, sachant la complexité et le volume des opérations requises.

Il y donc une double convergence entre les techniques probabilistes et celles d'intelligence artificielle: par le haut, à travers une interface avec une couche de haut niveau utilisant les notions d'objet et de contexte, et à l'intérieur même du modèle probabiliste, avec l'influence du discret, l'utilisation d'approximations, l'apparition de développement d'arbres d'hypothèses pour les problèmes de prédiction et de contrôle, l'utilisation d'heuristiques pour traiter les explosions combinatoires, et les techniques de décision à partir de représentations mixtes symboliques/probabilistes. Nous pouvons reprendre ici le schéma vu en 5.5.3.

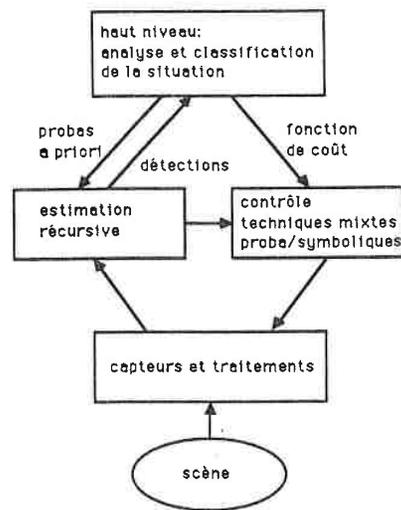


schéma d'un système de perception robotique en deux couches

Nous avons abordé ce problème de la complexification de l'environnement dans la dernière simulation (chapitre V), qui vise donc à regrouper un peu tous les aspects présentés dans ce mémoire, avec la présence d'objets de différents types dans un espace à deux dimensions, dotés d'une vitesse. Nous y avons également étudié l'intégration multisensorielle à travers la modélisation de deux types de capteur, radar et vision, et le problème de la prise de décision à partir d'hypothèses sur la situation et de leurs probabilités.

Le développement de robots de troisième génération (robots et véhicules autonomes) passera nécessairement par ce type de système d'intégration de la perception. L'utilisation conjointe de techniques issues de l'automatique et de techniques issues de l'intelligence artificielle reste donc un domaine important de la recherche en cours en robotique. On arrive alors à la fois:

- à un partage des tâches, les bouclages rapides et plutôt de bas niveau étant réalisés avec une classique réinjection de l'erreur, ou dans le cadre d'un modèle mathématique simple et précis, les contrôles utilisant des connaissances de haut niveau plus vagues, et étant en général plus désynchronisés du temps réel utilisant des techniques informatiques issues de l'intelligence artificielle
- à une fusion des techniques, avec des algorithmes de décision utilisant des représentations mixtes symboliques probabilistes.

Enfin, les techniques de décision du haut niveau elles-mêmes (systèmes de planification, logiques non monotones, classification du contexte.. [Lanusse]) n'ont pas été étudiées ici, mais

sont indispensables si l'on veut réaliser des robots évoluant dans des univers peu ou non contrôlés, de manière à ne jamais être pris au dépourvu par une situation non prévu par les modèles de bases, et à manipuler tout ce qui est connaissance vague et imprécise.

références

- Amat P.
Rapport de DEA, ETCA 85
- Anderson B.D.O., Moore J.B.
Optimal Filtering
Thomas Kailath Editor, Prentice-Hall, EE Series, 1979
- Bahl L.R., Jelinek F., Mercer R.L.
A maximum likelihood approach to continuous speech recognition
PAMI 5, no 2, march 83
- Bennis O., Huillet T., Monin A., Salut G.
Aspects non-linéaires en poursuite radar de cibles aériennes
Applied Control, Filtering and Signal Processing, IASTED Symposium, Genève, 1987
- Berger J.O.
Statistical decision theory
Springer Verlag, 1980
- Birmiwal K., Bar-Shalom Y.
Dual control guidance for simultaneous identification and interception
Automatica, vol 20, no 6, pp 737-749, 1984
- Blackman S. S.
Multiple-target tracking with radar applications
Artech House, 1986
- Bonneval A.
Photogrammétrie générale
collection scientifique IGN, Eyrolles
- Bouthermy P.
Estimation of edge motion based on local modeling
SPIE vol. 595, Computer Vision for Robots, Cannes, dec.85, pp 162-169
- Bouvet M.
Bayesian quickest signal detection in a discrete time observation
IEEE Trans. Aerosp. and Electron. Sys., vol AES-22, no 2, pp 170-176, march 1986
- Burg B., Zavidovique B.
Vers une application fondamentale de l'apprentissage: le contrôle de processus
2ème journée française de l'apprentissage, Chamrousse, mars 87
- Caines P.E., Chen H.F.
Optimal adaptive LQG control for systems with finite state parameters
IEEE Trans. on Automatic Control, vol AC-30, no 2, feb 85
- Carbonell J., Hood G.
"The world modeler project: objectives and simulator architecture"
Proc. Int. Machine Learning Workshop, 1985
- Cassou J.P., Millour C.
Perception multi-niveau et multi-traitement:
Application à la segmentation d'événements dans un écoulement turbulent
Rapport interne ETCA /CTME/OP-SP, 86
- Cheeseman P.
A method of computing generalized Bayesian probability values for expert systems
Proc. 8th IJCAI, Karlsruhe, W. Germany, pp 198-202, 1983

- Chen C.H.
Pattern Recognition and Signal Processing
NATO Series, Applied Sciences, Sijthoff & Noordhoff 1978
- Chiu S., Morley D., Martin J.
Sensor Data Fusion on a Parallel Processor
Proc. IEEE Conference on Robotics and Automation, pp1629,1633, San Francisco 1986
- Dickmanns E.D., Zapp A.
Guiding land vehicles along roadways by computer vision
congrès AFCET automatique, Toulouse 85
- Dubois D.
Modèles mathématiques de l'imprécis et de l'incertain en vue d'applications aux techniques d'aide à la décision
thèse INPG, 1983
- Dubois D., Prade H.
Théorie des possibilités: applications à la représentation des connaissances en informatique
éditions Masson, 1985
- Dubois D., Prade H.
Le traitement de l'imprécis et de l'incertain dans les systèmes experts
journées Science et Défense sur la robotique militaire, Toulouse 1985
- Durrant-Whyte H.
Consistent Integration and Propagation of Disparate Sensor Observations
Proc. IEEE Conference on Robotics and Automation, pp1464,1469, San Francisco 1986
- Durrant-Whyte H.
Integrating distributed sensor information, an application to a robot system coordinator
Proc. IEEE Conference on systems, man and cybernetics, p. 415, 1985
- El Jai A., Pritchard A.J.
Capteurs et actionneurs dans l'analyse des systèmes distribués
Masson 1986, Paris
- Faure A.
Perception et reconnaissance des formes
Editests, 1985
- Faurre P.
Notes d'optimisation
cours d'analyse numérique, Ecole Polytechnique, 1982
- Feldbaum A.A.
Optimal control systems
Academic Press, New York, 1965
- Fortmann T.E., Bar-Shalom Y., Scheffe M.
Multitarget tracking using joint probabilistic data association
IEEE J. Oceanic Eng., July 1983
- Foulloy L., Kechemair D., Burg B., Lamotte E., Zavidovique B.
A rule-based decision system for the robotization of metal laser cutting
IEEE conference on Robotics and Automation, St Louis, Missouri, USA, March 1987
- Garvey T.D., Fischler M.A.
Perceptual reasoning in a hostile environment
Proc. 1st Annual National Conference on AI, August 1980
- Garvey T.D., Fischler M.A.
The integration of multi-sensor for threat assessment
SRI, Stanford, 1980
- Ghallab M.
Optimisation de processus décisionnels pour la robotique
thèse d'état, LAAS/UPS 1982
- Gittins J.C., Jones D.M.
A dynamic allocation index for the sequential design of experiments
in J. Gani, K. Sarkadi & I. Vincze (eds): Colloquia Mathematica Societatis Janos Bolyai 1972, 9, progress in statistics, European Meeting of Statisticians, 241-266
North Holland, London
- Hansen C., Henderson T.C., Shilcrat E., Fai W.S.
Logical Sensor Specification
Proc. of SPIE Conference on Intelligent Robots, Nov 1983
- Hanson, Riseman
VISIONS
in Computer vision systems, NY Academic Press, 1978
- Kaufman A.
Méthodes et modèles de la recherche opérationnelle
Dunod 64, p. 221
- Kaufman A.
Introduction à la théorie des sous-ensembles flous
Masson 73
- Kent E.W., Albus J.S.
Sensored World Models as interfaces between control systems and sensory data
NBS, Washington, 84, Robotica vol 2 pp 17-25
- Kumar P.R.
A survey of some results in stochastic adaptive control
SIAM J. Contr. Optimiz. vol 23, 1985
- Kumar P.R., Varaiya P.
Stochastic Systems: Estimation, Identification and Control
EECS 223 notebook, 86, Univ. of Cal. Berkeley
- Lanusse A.
thèse d'état à venir, ETCA, Arcueil
- Lanusse A., Merlo X., Zavidovique B.
Modèles d'intégration et de reconfiguration multisensorielle en robotique
Proc. COGNITIVA 87, Paris
- Lanusse A., Merlo X., Desnoyer J.L.
Etude et développement d'algorithmes coopératifs de reconnaissance des formes pour un système de perception multicapteur
Rapport final contrat DRET 86/1015
- Lefèvre P.
Estimation récurrente optimale, filtre de Kalman
cours ENSTA, 1970
- Lindley L.V.
Scoring rules and the inevitability of probability
International Statistical Review, 50, 1-26, 1982
- Marroquin
Probabilistic solution of inverse problems
Ph D Thesis, MIT, September 85

Merlo X., Lanusse A., Zavidovique B.
Contrôle optimal d'un système de perception robotique
Applied Control, Filtering and Signal Processing, IASTED Symposium, Genève, 1987

Métivier M., Neveu J.
Probabilités
cours Ecole Polytechnique, 1980

Millour C., Lanusse A.
Intégration de mécanismes préattentifs en analyse par vision de scènes dynamiques
2ème atelier TIPI, CNRS 88

Minsky M.
A framework for representing knowledge
The psychology of computer vision
McGraw Hill, Computer Science Series, PH Winston, ed. 1975

Moravec K., Nasburg R.
Multiple sensor classification for air defence
TI Engineering Journal, may-june 1985

Nii H.P., Feigenbaum E.A., Anton J.J., Rockmore A.J.
Signal-to-Symbol transformation : HASP / SIAP Case Study
the AI Magazine, spring 82

Piaget J., Inhelder B.
La représentation de l'espace chez l'enfant
1948, édition PUF 1981

Reid D.B.
The application of multiple target tracking theory to ocean surveillance
IEEE 1979

Rives P., Espiau B.
Une méthode d'estimation récursive en boucle fermée de primitives 3D utilisant une camera
mobile
Proceedings COGNITIVA 87, Paris

Rives P., Hegron G.
Design of a simulation tool for robots using moving vision sensors
Int. NATO Workshop Languages for sensor-based control in Robotics
Castelvecchio, Italy, sept. 86

Rives P., Marce L.
Use of moving vision sensors to control robots in an unknown universe
Proceedings ROVISEC 5th, Amsterdam, oct. 85

Sallantin J.
Représentation d'observations dans le contexte de la théorie de l'information
Thèse d'état, université Paris VI, 1979

Shafer G.
A mathematical theory of evidence
Princeton University Press, Princeton N.J. 1976

Spiegelhalter D. J.
Probabilistic reasoning in predictive expert systems
Uncertainty in artificial intelligence,
LN Kanal & JF Lemmer editors,
Elsevier Science Publishers, 1986

Sworder D.D.
Optimal adaptive control
New York, Academic, 1965

Sworder D.D., Chou D.S.
Feedforward / Feedback controls in a noisy environment
IEEE Trans on Sys., Man and Cyb., vol SMC-16, no 4, july-august 86

Tse E., Bar-Shalom Y.
Actively adaptative methods for stochastic systems
Proc. 18th IEEE Conf. Decision Contr., 1979

Wald A.
Statistical decision functions
Wiley, New York 1950

Willsky A.S., Jones H.L.
A generalized likelihood approach to the detection and estimation of jumps in linear systems
IEEE Trans. on Automatic Control, vol AC-21, no 1, feb 76

Wolfe J.
Les processus visuels cachés
La perception visuelle, Bibliothèque Pour la Science, 1984

Zadeh L.A.
Fuzzy sets as a basis for a theory of possibility
Fuzzy Sets and Systems 1, 3-28, 1978

Zadeh L.A.
Fuzzy sets versus probability
Proc. IEEE, 68, 421, 1980

Zavidovique B.
communications privées, 1985-1987

Zavidovique B.
Méthodes d'intégration avancée de dispositifs robotiques
Rapport final contrat DRET 85.34.813.00.470.75.01, 1987

